2020-08-29

Feasibility analysis of using the hp-adaptive Radau pseudospectral method for minimum-effort collision-free docking operations of AUV

Yazdani, A.M.; Sammut, K.; Yakimenko, O.A.; Lammas, A.

Elsevier

Yazdani, Amir Mehdi, et al. "Feasibility analysis of using the hp-adaptive Radau pseudospectral method for minimum-effort collision-free docking operations of AUV." Robotics and Autonomous Systems 133 (2020): 103641.
http://hdl.handle.net/10945/68038

Robotics and Autonomous Systems 133 (2020) 103641

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Feasibility analysis of using the hp-adaptive Radau pseudospectral method for minimum-effort collision-free docking operations of AUV

A.M. Yazdani [a,b], K. Sammut [b], O.A. Yakimenko [c,*], A. Lammas [b]

[a] College of Science, Health, Engineering and Education, Murdoch University, Perth, WA 6150, Australia
[b] Centre for Maritime Engineering, Control and Imaging, College of Science and Engineering, Flinders University, Adelaide, SA 5042, Australia
[c] Department of Systems Engineering/Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA 93943, United States of America

ARTICLE INFO

Article history:
Received 4 March 2020
Received in revised form 28 June 2020
Accepted 28 August 2020
Available online 29 August 2020

Keywords:
Underwater docking
Optimal trajectory generation
Two-point boundary-value problem
hp-adaptive Radau method

ABSTRACT

This paper continues the previous effort on the development of a trajectory generation platform that assures minimum-control expenditure and collision-free manoeuvre of a torpedo-shaped autonomous underwater vehicle (AUV) into a funnel-shaped stationary docking station (DS). The earlier-developed guidance system was based on the Inverse Dynamics in the Virtual Domain (IDVD) method accounting for AUV's dynamics and producing a smooth trackable trajectory, thus guaranteeing safe arrival to DS. The optimality of the real-time generated solutions has been assessed via comparing them with the Legendre–Gauss–Lobatto pseudo-spectral (PS) method solutions that could only be obtained off-line. This paper explores a possibility of employing a more advanced hp-adaptive Radau (hp-AR) PS method for the same Hamiltonian two-point boundary-value problem. The considered approach explicitly encapsulates all realistic vehicular and environmental constraints such as the AUV's dynamics, ocean current disturbances, no-fly zones, and DS pose while minimizing the vehicle's controls expenditure and permitting precise manoeuvring into DS. The performance evaluation of the hp-AR PS based optimization routine is carried out through extensive software-in-the-loop simulations. For completeness, computational effectiveness and solution optimality of the trajectory generator engine based on the hp-AR method is compared with two other well-known PS methods based on Legendre and Chebyshev polynomial approximation. The results of this study show the superb performance of the hp-AR method-based trajectory generator among all other PS methods and a possibility of using it along with IDVD in the real-time implementation.

Published by Elsevier B.V.

1. Introduction

In the recent years, autonomous underwater vehicles (AUVs) became increasingly popular for survey and exploration missions. A plethora of AUV application studies such as mine detection, environmental profiling, bathymetric mapping, surveillance, and reconnaissance have been documented in the literature [1–4]. The expanding mission scope for AUVs is however limited by the propulsion system, specifically, battery capacity and hotel-load consumption, shortening mission endurance. While there have been some significant improvements in battery energy density, the capacity to conduct long-term missions, especially with power hungry payloads, over many days to months has still not been realized. This restriction imposes an extra cost of manned deployment and recovery for the purpose of battery recharging to extend mission duration. The use of a submerged docking station (DS) that can permit battery recharging and data uploading, is one possible means to enable persistence while reducing operation costs and hazards [5,6].

The use of DS implies an ability to autonomously locate it and guide AUV into the DS entrance. An AUV docking guidance system should take into account all operational constraints including the DS entrance aperture, which imposes a restriction on a vehicle's final approach angle, the thruster saturation margins, and kinematic constraints limiting AUV's radius and rate of turn. Additionally, the candidate guidance system should be able to accommodate various uncertainties including the initial pose of the vehicle with respect to DS, and environment variability in terms of current disturbances and static/dynamic no-fly (NF) zones (obstacles).

A survey of different approaches to accomplish an underwater docking as implemented in the existing systems is provided in [7]. To name a few, there are quite a few studies that benefited from the relative ease of implementation of the point-to-point guidance for docking operations [7–10]. However, their performance

* Corresponding author.
E-mail address: OAYAKIME@NPS.EDU (O.A. Yakimenko).

https://doi.org/10.1016/j.robot.2020.103641
0921-8890/Published by Elsevier B.V.

is confined in a generating simple point-to-point trajectory in a usually controlled conditions and without considering vehicle's dynamics and operating environment. Less attention has been devoted to the development of AUV guidance laws based on the optimal control theory. That includes the development of an optimal trajectory generator framework that enabled AUV to perform minimum-time or minimum-effort collision-free underwater manoeuvres [11] and a constrained energy-optimal solution employing realistic constraints of AUV thrusters [12]. In [13], an analytical time optimal trajectory solution for depth control was achieved in a closed-form using two-degree-of-freedom dynamics. In [14], an analytical relationship between the number of AUV thrusters and energy consumption was used to develop a minimum-control-expenditure manoeuvre, and in [15], the optimal control theory was employed to develop a minimum-time trajectory for a fully actuated AUV subject to a constraint on the input force. For a larger-scale problem dealing with long-range guidance around islands near the undersea volcano rather than short-range docking guidance, [16,17] implemented a concept of quasi-optimal navigation based on the analytically-derived guidance law for a three degree-of-freedom (3-DoF) AUV model with a single control input followed by varying the initial heading to reach the intended point of destination. In the case of unexpected change in the current flow, the AUV switches to proportional navigation or straight-line tracking.

The reason that the optimal control theory has lesser practical realizations is that obtaining an analytical solution for a generic optimal control problem (OCP) involving higher-order dynamics and feasible constraints is very difficult if not impossible. Instead, OCP is usually solved numerically employing one of direct methods of calculus of variations transcribing the original OCP into a nonlinear programming problem (NLP). The latter is then solved by parametric approximation of the states and/or controls with a subsequent parameter optimization conducted using either gradient-based techniques like sequential quadratic programming or gradient-free evolutionary-based techniques like genetic and simulated annealing algorithms [18].

Following the aforementioned approach, the authors have developed a robust docking guidance system based on the Inverse Dynamics in the Virtual Domain (IDVD) method that considers a higher-DoF AUV model and thus allows taking AUV's dynamics into account while generating quasi-optimal solutions [19]. Trackability of these trajectories, generated in real time, was verified using a high-fidelity software-in-the-loop AUV simulation platform (SITLSP) [20]. While IDVD does provide a real-time suboptimal solution, [19] has also made an attempt to compare it with the off-line solution provided by one of the direct orthogonal collocation aka pseudospectral (PS) methods.

PS methods have been widely utilized in the past two decades to solve a broad range of trajectory optimization problems. That includes both aerospace applications [21–25] and (more rarely) underwater trajectory optimization problems [26–29]. The essence of PS methods is in the use of orthogonal polynomials, like the Lagrange and Chebyshev polynomials (referred to as LPS and CPS methods, respectively) for the aforementioned parametric approximation of a solution at a set of discrete (collocation) points resulting from applying the Gaussian quadrature rule [30–33]. The optimization routine imposes the path and boundary constraints on the collocation points and minimizes the sum of the squared collocation residuals at midpoints formed as a difference between the derivative of approximating polynomial and actual vehicle dynamics described by a set of differential equations. The Karush–Kuhn–Tucker optimality conditions are tied to costate variables [30,4].

Typically, PS methods require substantial computational resources which precludes using them in real-time applications [30,33–35]. Moreover, these methods are known to exhibit slow convergence or even divergence for non-smooth OCPs because of necessity to use the higher-order approximating polynomials. The problem aggravates with increasing the complexity of the model (adding more state variables). It should be noted that to initialize the off-line optimal solution search in [19], the IDVD solution was used to produce an initial guess. Due to the IDVD quasi-optimal solution closeness to the truly optimal solution, finding the latter one using the PS method was always successful. Still, it took over two orders of magnitude more time required to obtain a solution as compared to the manoeuvre time itself. Also, the PS method struggled to obtain a solution in the case of a cluttered operating environment.

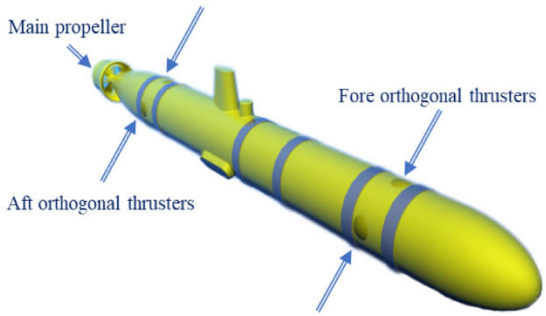

Fig. 1. Flinders AUV.

This paper explores the usage of yet another, more advanced PS method to overcome aforementioned pitfalls. It employs the hp-adaptive Radau (hp-AR) PS method exploiting multi-interval approximations with the lower-order polynomials [32,21,25]. While the IDVD solution is still used as an initial guess, the hp-AR PS method is capable of handling OCP much more efficiently, requiring less computational efforts and capable of easily handling state constraints by adjusting the locations and number of computational nodes (collocation points). The remainder of this paper is organized as follows. Section 2 briefly introduces the model of the underwater docking mission. More details on this model, along with the corresponding Hamiltonian boundary-value problem (HBVP) formulation can be found in [19]. Section 3 provides a brief description of the hp-AR method followed by Section 4, where performance of the hp-AR method-based guidance system is assessed against two realistic benchmark scenarios. Section 5 reviews the results of assessing the effectiveness of realistic representation of underwater docking operations with the hp-AR method-based optimization from the standpoint of trackability of optimal trajectories generated using the SITLSP described in [19,20]. Using Monte Carlo simulations, Section 5 also addresses the issues of computational effectiveness (capability of working in real time), solution optimality and robustness. For completeness, the hp-AR method solutions are compared with those of LPS and CPS. The paper ends with conclusions.

2. Underwater docking problem model

Fig. 1 shows the Flinders AUV, which is currently under development. It features a common torpedo-shaped configuration with a length of about 1.2 m and a diameter of 0.2 m. The forward motion (of up to 4 knots) is provided by a main propeller, two lateral and two vertical thrusters are responsible for the yaw and depth control, respectively. Such an arrangement results in decoupling of the controlled yaw and heave motion from the roll and pitch motion.

A stationary funnel-shaped DS considered in this research (Fig. 2) allows unidirectional docking approach at a very low

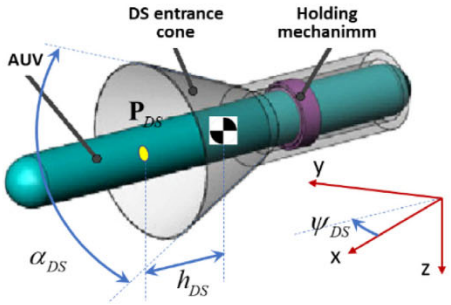

Fig. 2. AUV docked to DS.

speed with a narrow bound of variation in a yaw direction. The position of a DS entrance point in the North-East-Down coordinate frame $\{n\}$ is defined by three Cartesian coordinates $\mathbf{P}_{DS} = [x_{DS}, y_{DS}, z_{DS}]^T$. The origin of $\{n\}$ is set at the surface, so that z represents the vertical distance from the surface down, i.e. depth. The DS orientation in the horizontal plane relative to North is defined by angle ψ_{DS}, entrance cone apex angle — by α_{DS}, and DS funnel depth (distance from the DS funnel entrance to the centre of gravity (CG) of a docked AUV) — by h_{DS}. The specific parameters utilized in this paper were taken from [36] and are the same as in [19].

Typically, the docking operation is broken into two phases. In the first phase, called the *homing* phase, AUV manoeuvres to position itself approximately 10 m . . . 20 m away from DS in rough alignment with its entrance cone centreline, pointed into the funnel. The second phase, called the *docking* phase, comprises a more-or-less straight-in approach with small corrections to maintain alignment with the centreline. In this research both phases are combined into one so that AUV's starting position and orientation with respect to DS may be quite arbitrary.

A complete 6-DoF model of a rigid-body AUV is composed of three groups of equations. Three differential equations describe the well-known kinematics of translational motion

$$\dot{\mathbf{P}} = \mathbf{R}_b^n \mathbf{V} + \mathbf{C} \tag{1}$$

where $\mathbf{P} = [x, y, z]^T$ is a vector of Cartesian coordinates of AUV's CG in $\{n\}$, $\mathbf{V} = [u, v, w]^T$ is the velocity vector in the body coordinate frame $\{b\}$ (with u, v and w being its surge, sway and heave components), $\mathbf{R}_b^n$ represents a rotation matrix (from $\{b\}$ to $\{n\}$) and $\mathbf{C} = [c_x, c_y, c_z]^T$ is the current vector.

Another three equations describe kinematics of rotational motion (angular velocity transformation)

$$\dot{\boldsymbol{\Theta}} = \mathbf{T}_b^n \boldsymbol{\omega} \tag{2}$$

Here $\boldsymbol{\Theta} = [\phi, \vartheta, \psi]^T$ is the Euler angle vector composed of roll, ϕ pitch ϑ and yaw ψ (describing orientation of $\{b\}$ with respect to $\{n\}$), $\boldsymbol{\omega} = [p, q, r]^T$ is an angular velocity vector in $\{b\}$ and $\mathbf{T}_b^n$ represents the corresponding transformation matrix.

Six differential equations describing the dynamics of a body immersed in a viscous fluid and developed on the grounds of hydrodynamic modelling theory [37,38] are often represented as [39,40]

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \mathbf{w} \tag{3}$$

where $\mathbf{v} = [\mathbf{V}^T, \boldsymbol{\omega}^T]^T$, $\boldsymbol{\eta} = [\mathbf{P}^T, \boldsymbol{\Theta}^T]^T$, $\mathbf{M}$ is the symmetric positive-definite system inertia matrix (including added masses and inertia moments), $\mathbf{C}(\mathbf{v})$ is the Coriolis–centripetal matrix (including added masses as well), $\mathbf{D}(\mathbf{v})$ is the damping matrix and $\mathbf{g}(\boldsymbol{\eta})$ is the vector of gravitational/buoyancy forces and moments. The right-hand side of Eq. (3) includes all the forces and moments that have effect on the motion of a vehicle combined into the control input vector $\boldsymbol{\tau}$ and the environmental disturbances vector $\mathbf{w}$.

A complete set of these 12 equations (Eqs. (1)–(3)) written in the scalar form can be found in [41–43]. Moreover, a complete set of hydrodynamic coefficients is given in [41] and [44,45] as applied to a design of a controller for the Aeris prolong-cuboid AUV and the REMUS torpedo-shaped AUV, respectively. The complete nonlinear model is typically used in software-in-the-loop and hardware-in-the-loop simulations. In particular, the SITLSP, presented in [19,20] and used in this study to verify feasibility of the guidance commands tracking, utilizes a full nonlinear model of [44].

For the purpose of a controller design, a complete set of Eqs. (1)–(3) is never used. Instead, a full model is being simplified, decoupled and linearized [42,44,46]. In particular, for addressing the guidance problems, researchers often use a subset of kinematic equations (Eq. (1)) only. Moreover, most of the time, researchers limit their studies to the planar motion, so that the driving equations in the horizontal plane become as simple as

$$\begin{aligned} \dot{x} &= u\cos(\psi) + c_x \\ \dot{y} &= u\sin(\psi) + c_y \end{aligned} \tag{4}$$

where c_x and c_y are the x- and y-components of the perceived current velocity vector. In Eq. (4), the control inputs are u and ψ (most often — just ψ with $u = fix$, e.g., [16,17]). Depending on the design of AUV, i.e. its control scheme (holonomic vs. nonholonomic), these equations may have one additional term (adding the sway motion) (e.g., [47,48]).

Adding the third equation describing a vertical motion also depends on the control scheme. For the specific AUV [19] the vertical motion is decoupled, so that Eq. (4) gets complimented with

$$\dot{z} = w + c_z \tag{5}$$

where w serves as an additional control input, and c_z represents the z-component of the perceived current velocity vector.

Accounting for a specific control scheme [19], which results in no roll and pitch motion

$$\phi = \vartheta = p = q = 0 \tag{6}$$

rotational kinematics of Eq. (2) reduces to

$$\dot{\psi} = r \tag{7}$$

In this case, r becomes a new control input (instead of ψ).

Having Eqs. (4), (5), (7) is usually sufficient to address any guidance problem (e.g., [16,17,42,47,47]). In this case, an optimal or suboptimal solution (obtained via implementing some heuristics or optimization procedure), given by the time histories $u(t)$, $w(t)$, and $r(t)$, is converted to a set of waypoints x_i, y_i, z_i, u_i, $i = 1, 2, \ldots, N$ that a typical autopilot understands and attempts to execute. The problem, however, is that since vehicle's dynamics was not accounted for at the guidance generation stage, some of the aforementioned waypoints are often not achievable, which results in an actual trajectory being somewhat different with respect to optimized guidance laws.

That is why [19] proposed a novel approach allowing to account for vehicle's dynamics at the stage of the optimal guidance law generation. No doubt, that involving more differential equations results in an additional computational burden. Employing the optimal control theory faces curse of dimensionality and convergence problems. In [19] these deficiencies were overcome by implementing the IDVD method relying on a differential flatness property of an underlying system of differential equations which allows finding suboptimal control solution via inverting vehicle's dynamics and operating in the virtual domain. It was shown that

this solution can be used as is, or as an initial guess for a more sophisticated optimal control routine, namely LGL PS [19].

Accounting for Eq. (6), pertinent to the control architecture of the Flinders AUV (Fig. 1), the dynamic equations (3) are reduced to

$$\begin{aligned}(m-X_{\dot u})\dot u - X_u u - X_{u|u|}\,|u|\,u - X_{rr}r^2 &= T_u\\ (m-Z_{\dot w})\dot w - Z_w w - Z_{w|w|}\,|w|\,w &= T_w\\ (I_z-N_{\dot r})\dot r - N_r r - N_{r|r|}\,|r|\,r - N_{ur}ur &= T_r\end{aligned} \tag{8}$$

In these equations, m and I_z are the AUV's mass and moment of inertia around the z-axis of $\{b\}$; $X_{\dot u}$, $Z_{\dot w}$, and $N_{\dot r}$ represent the added mass and inertia terms; X_{rr} and N_{ur} are the added mass cross terms; X_u, Z_w, N_r, $X_{u|u|}$, $Z_{w|w|}$ and $N_{r|r|}$ are the linear and quadratic drag terms; T_u, T_w, T_r — surge force, heave force, and yaw moment, respectively. While these parameters are unique for each particular AUV, analysis shows that the X_{rr} and N_{ur} terms in Eq. (8) are smaller than others and can be neglected. The linear and quadratic drag terms may be prevailing at different speeds [41–45,49–58], added mass terms $X_{\dot u}$, $Z_{\dot w}$, and $N_{\dot r}$, specifically the latter two, depend on AUV's geometry and may be substantial compared to m and I_z (on the order of 10% to 100% of the corresponding parameters in the rigid body mass matrix [44,59–62].

With Eq. (8), the control vector now becomes $\mathbf{U} = [T_u, T_w, T_r]^T$. The turn rate and controls are usually bounded

$$|r| \le r_{\max},\ |T_u| \le T_{u\max},\ |T_w| \le T_{w\max},\ |T_r| \le T_{r\max} \tag{9}$$

While Flinders AUV parameter identification (ID) is still in progress, [19] used the values of these parameters as defined for the REMUS AUV, reported in [44]. Both [19] and this paper present a general idea on how AUV's dynamics can be incorporated into the optimal guidance generation, and from this standpoint the exact values of coefficients in Eq. (8) are not so important (they can be corrected later, after the Flinders AUV ID is completed). To be consistent with the results of [19], this study uses the same model as in [19,56,63,64]. Once the optimal guidance laws based on Eqs. (4), (5), (7), (8) are developed, they are tested on a SITLSP [20], that runs a full nonlinear 6-DoF model (1)–(3) to see whether these guidance laws are realizable. Validation procedure also includes varying the coefficients of Eq. (8) against the corresponding coefficients of the full model of SITLSP to prove robustness of the proposed approach.

Again, for the sake of apples-to-apples comparison, this paper utilizes the same model as in [19] and addresses the fixed-time minimum-thrust optimal control problem minimizing the normalized cost function

$$J = \frac{1}{T_{u\max}^2}\int_{t_0}^{t_f}(T_u^2 + T_w^2 + T_r^2)dt \tag{10}$$

As such, this study utilizes the same expressions for the Hamiltonian and costates as in [19]. The two-point boundary value problem assumes departing from some starting point defined by the state vector

$$\mathbf{X}_0 = [x_0, y_0, z_0, \psi_0, u_0, w_0, r_0]^T \tag{11}$$

to the final point

$$\mathbf{X}_f = [x_f, y_f, z_f, \psi_f, u_f, w_f, r_f]^T \tag{12}$$

defined by location of the DS entrance point, $\mathbf{P}_{DS}$, and its orientation ψ_{DS} (Fig. 2), so that

$$\psi_f = \psi_{DS} + \pi \tag{13}$$

In all simulations shown in this paper, the current flow is represented by the constant vector $\mathbf{C} = const$ (see Eq. (1)), reflecting the best knowledge of the current environment. This is an assumption that all short-range short-duration docking operations are based upon (e.g. [36]). The guidance system developed in [19] allows to easily incorporate any spatial or temporal variations if they are available or recompute the trajectory with the updated currents if/when they become available during docking. In this sense, it is similar to the approach outlined in [16,17] deriving the long-range long-duration optimal heading sequences within a finite computational time, provided that the current velocity in the navigation region is known a priori.

The optimal solutions obtained in this paper using the hp-AR method are also compared with the ones obtained using the LPS and CPS methods. All three PS methods are first compared for the obstacle-free environment, and then — for the cluttered environment with multiple NF zone path constraints of the form

$$(x-x_{NF})^2 + (y-y_{NF})^2 + (z-z_{NF})^2 \ge r_{NF}^2 \tag{14}$$

where the vector $\mathbf{P}_{NF} = [x_{NF}, y_{NF}, z_{NF}]^T$ denotes the centre of the NF zone with the r_{NF} radius.

The next section provides a brief description of the hp-AR method employed to numerically resolve the Hamiltonian boundary-value problem (HBVP) based on the model outlined in this section (and fully described in [19].

3. Essence of the hp-adaptive radau method

This section presents the mathematical foundations of the Legendre–Gauss–Radau (LGR) collocation approach to numerically resolve HBVP based on the model presented in Section 2 (see [19] for more details), followed by a discussion of adaptively refining the mesh to improve accuracy and computational efficiency.

3.1. The LGR collocation

The collocation points mesh is created by dividing the time interval $[t_0; t_f]$ into K subintervals, $t \in [t_{k-1}; t_k]$, $k = 1, 2, \ldots, K$, so that $0 = t_0 < t_1 < \cdots < t_K = t_f$. Each subinterval is then scaled to $\tau^{(k)} \in [-1; 1]$ by using the affine transformation $\tau^{(k)} = (2t - t_k + t_{k+1})(t_k - t_{k-1})^{-1}$, $t_{k-1} \le t \le t_k$. The state vector for each subinterval is approximated as [21,65,66]

$$\begin{aligned}\mathbf{X}^{(k)}(\tau^{(k)}) &= \sum_{i=1}^{N_k+1}\mathbf{X}^{(k)}(\tau_i^{(k)})\varphi_{N_k,i}^{(k)}(\tau^{(k)}),\\ \varphi_{N_k,i}^{(k)}(\tau^{(k)}) &= \prod_{\substack{j=1\\ j\ne i}}^{N_k+1}\frac{\tau^{(k)} - \tau_j^{(k)}}{\tau_i^{(k)} - \tau_j^{(k)}}\end{aligned} \tag{15}$$

where $\tau_1^{(k)}, \tau_2^{(k)}, \ldots, \tau_{N_k}^{(k)}$ are the LGR collocation points — the roots of the N_kth-degree Lagrange interpolating polynomial for each kth subinterval (N_k may vary from subinterval to subinterval). Since LGR points lie in the half open subinterval $\tau^{(k)} \in [-1; 1)$, approximation (15) includes the terminal point, denoted as point N_k+1, i.e. $\tau_{N_k+1}^{(k)} = 1$. The terminal point of the Kth subinterval is denoted by $\tau_{N_K+1}^{(K)}$. The collection of points $\tau_1^{(k)}, \ldots, \tau_{N_k}^{(k)}$, $k = 1, 2, \ldots, K$ constitute a set of $\sum_{k=1}^{K} N_k$ global collocation points at which the state equations (4), (5), (7), (8) are enforced. The algorithm to vary the values of K and N_k, $k = 1, 2, \ldots, K$ (h and p adaptation, respectively) is discussed in the next section.

The control variables in all but the last mesh interval $k \in [1; K-1]$ are defined in a similar manner

$$\begin{aligned}\mathbf{U}^{(k)}(\tau^{(k)}) &= \sum_{i=1}^{N_k+1} \mathbf{U}^{(k)}(\tau_i^{(k)})\varphi_{N_k,i}^{(k)}(\tau^{(k)}),\\ \varphi_{N_k,i}^{(k)}(\tau^{(k)}) &= \prod_{\substack{j=1\\ j\neq i}}^{N_k+1} \frac{\tau^{(k)}-\tau_j^{(k)}}{\tau_i^{(k)}-\tau_j^{(k)}}\end{aligned} \tag{16}$$

For the last subinterval though, the (N_k-1)th-degree Lagrange interpolating polynomial is used (because the final point of subinterval K, corresponding to $t=t_f$, is not collocated [21,65]

$$\begin{aligned}\mathbf{U}^{(K)}(\tau^{(k)}) &= \sum_{i=1}^{N_k} \mathbf{U}^{(K)}(\tau_i^{(K)})\hat{\varphi}_{N_k,i}^{(K)}(\tau^{(K)}),\\ \hat{\varphi}_{N_k,i}^{(K)}(\tau^{(K)}) &= \prod_{\substack{j=1\\ j\neq i}}^{N_k} \frac{\tau^{(K)}-\tau_j^{(K)}}{\tau_i^{(K)}-\tau_j^{(K)}}\end{aligned} \tag{17}$$

The residuals are formed by differentiating $\mathbf{X}^{(k)}(\tau^{(k)})$ with respect to $\tau^{(k)}$ and then collocating state equations (4), (5), (7), (8) at each of the N_k LGR points of each subinterval k [21,66]

$$\begin{aligned}&\sum_{i=1}^{N_k+1} \mathbf{D}_{hi}^{(k)}\mathbf{X}^{(k)}(\tau_i^{(k)}) - \frac{t_k-t_{k-1}}{2}\mathbf{f}(\mathbf{X}^{(k)}(\tau_i^{(k)}), \mathbf{U}^{(k)}(\tau_i^{(k)}), \tau_i^{(k)}, t_{k-1}, t_k)\\ &= \mathbf{0}, \text{ for } h=1,2,\ldots,N_k\end{aligned} \tag{18}$$

where $\mathbf{D}_{hi}^{(k)}$ is the N_k–by–(N_k+1) Radau pseudospectral differentiation matrix, composed of elements $d_{hi}^{(k)}$ defined as [30]

$$d_{hi}^{(k)} = \dot{\varphi}_i(\tau_h^{(k)}) = \begin{cases} \dfrac{\dot{g}(\tau_h^{(k)})}{(\tau_h^{(k)}-\tau_i^{(k)})\dot{g}(\tau_i^{(k)})} & \text{if } h\neq i\\ \dfrac{\ddot{g}(\tau_i^{(k)})}{2\dot{g}(\tau_i^{(k)})} & \text{if } h=i\end{cases} \tag{19}$$

and

$$g(\tau_h^{(k)}) = (1+\tau_h^{(k)})(P_{N_k+1}(\tau_h^{(k)}) - P_{N_k}(\tau_h^{(k)})) \tag{20}$$

(here $P_{N_k}(\tau_h^{(k)})$ and $P_{N_k+1}(\tau_h^{(k)})$ denote the N_kth- and (N_k+1)th-degree Legendre polynomials).

The boundary conditions are imposed as equality constraints on the starting point of the first subinterval $(\mathbf{X}_1^{(1)}, t_0)$ and the terminal point of the Kth subinterval $(\mathbf{X}_{N_k+1}^{(K)}, t_f)$. Meanwhile, the path constraints in each subinterval are satisfied at N_k LGR points. The cost function (10) is approximated by the quadrature rule with the LGR weights [30]

$$w_i^{(k)} = \begin{cases} 2N_k^{-2} & \text{for } i=1\\ (1-\tau_i^{(k)})^{-1}(\dot{P}_{N_k-1}(\tau_i^{(k)}))^{-2} & \text{for } i=2,3,\ldots,N_k\end{cases} \tag{21}$$

The Radau method offers its own costate mapping function to provide equivalent continuous costates and Hamiltonian trajectories for the underlying HBVP. The mathematical details that provide a complete discretization of the continuous necessary conditions for the Radau method can be found in [31–33].

3.2. The hp adaptation

To mitigate the difficulties associated with an extremely slow convergence rate (or even divergence) when addressing non-smooth OCPs, even when employing the high-degree basis functions [30,65,66], the hp-adaptive algorithm (as applied to the Radau multi-interval method described in Section 3.2), assumes either increasing the degree of approximating polynomial (increasing the number of collocation points) or subdividing the original subintervals onto the smaller-size subintervals when appropriate [67,68].

The algorithm is initialized by configuring a coarse computational mesh comprised by the even-size subintervals with the same degree of approximating polynomial in each subinterval. Then, OCP is transcribed into NLP problem using the approach presented in Section 3.1).

For each subinterval $k=1,2,\ldots,K$, the approximation error in satisfying discretized dynamics and constraints is estimated at midpoints of the collocation points $\overline{\tau}_l^{(k)} = 0.5(\tau_l^{(k)} + \tau_{l+1}^{(k)})$, $l=1,2,\ldots,N_k-1$ [21]

$$\left|\dot{\mathbf{X}}^{(k)}(\overline{\tau}_l^{(k)}) - \frac{t_k-t_{k-1}}{2}\mathbf{f}(\mathbf{X}^{(k)}(\overline{\tau}_l^{(k)}), \mathbf{U}^{(k)}(\overline{\tau}_l^{(k)}), \overline{\tau}_l^{(k)}, t_{k-1}, t_k)\right| = \mathbf{e}_l^{(k)} \tag{22}$$

$$\left|\mathbf{g}^{(k)}(\mathbf{X}^{(k)}(\overline{\tau}_l^{(k)}), \mathbf{U}^{(k)}(\overline{\tau}_l^{(k)}), \overline{\tau}_l^{(k)}, t_{k-1}, t_k)\right| = \mathbf{b}_l^{(k)} \tag{23}$$

If the accuracy of every element in $\mathbf{e}_l^{(k)}$ and $\mathbf{b}_l^{(k)}$ is within the desired tolerance ε, meaning $e_{l,i}^{(k)} \leq \varepsilon$, for all $i=1,2,\ldots,n$, and $b_{l,i}^{(k)} \leq \varepsilon$, for all $i=1,2,\ldots,c$, no mesh refinement is needed. Otherwise, either the increase of the degree of the corresponding approximating polynomial in the subinterval in question or reduction of the size of this interval (introduction of several smaller-size intervals) is required, (p and h adaptation, respectively). This decision is based on evaluating a relative curvature spread as follows.

The curvature at midpoints $\overline{\tau}_l^{(k)}$ can be evaluated using the well-known relationship

$$\kappa_{l,i}^{(k)} = \left|\ddot{\mathrm{X}}_i^{(k)}(\overline{\tau}_l^{(k)})\left(1+\dot{\mathrm{X}}_i^{(k)}(\overline{\tau}_l^{(k)})^2\right)^{-3/2}\right| \tag{24}$$

In Eq. (24), $\mathrm{X}_i^{(k)}(\overline{\tau}_l^{(k)})$, $i=1,2,\ldots,n$, $l=1,2,\ldots,N_k-1$ denote the ith component of the state vector approximation at midpoint l within the mesh subinterval k that does not pass the error test for $\mathbf{e}_l^{(k)}$ (Eq. (22)) or $\mathbf{b}_l^{(k)}$ (Eq. (23)). The entire kth interval can then be characterized by the maximum value of $\kappa_{l,i}^{(k)}$ among all midpoints l and state vector components i, $\kappa_{\max}^{(k)}$. Moreover, for the state vector component I, the value of $\kappa_{\max}^{(k)}$ corresponds to, the relative curvature variance can be defined as

$$r^{(k)} = \frac{\kappa_{\max}^{(k)}}{\overline{\kappa}^{(k)}}, \text{ where } \overline{\kappa}^{(k)} = \frac{1}{N_k-1}\sum_{l=1}^{N_k-1}\kappa_{l,I}^{(k)} \tag{25}$$

If $r^{(k)}$ happens to be less than some user-defined maximum limit $r_{\max}$, the degree of polynomial (the number of collocation points) should be increased. In this case, the polynomial degree increase is accomplished according to [17,65,66]

$$N_k^+ = N_k^- + \mathrm{ceil}(\log_{10}(e_{\max}^{(k)}) - \log_{10}(\varepsilon)) + A \tag{26}$$

where N_k^+ and N_k^- denote the new and original polynomial degree, $e_{\max}^{(k)}$ denotes the maximum midpoint approximation error (among all $e_{l,i}^{(k)}$, $i=1,2,\ldots,n$ and $b_{l,i}^{(k)} \leq \varepsilon$, $i=1,2,\ldots,c$, while varying $l=1,2,\ldots,N_k-1$), A is an arbitrary integer number for adjusting the growth of the number of collocation points in the kth subinterval, and "ceil" denotes the round towards plus infinity operator.

If, however, $r^{(k)}$ in Eq. (25) is significantly larger than $r_{\max}$, the mesh refinement should be applied by subdividing the original subinterval k onto few more subintervals according [21,65,66]

$$n_k = B\mathrm{ceil}(\log_{10}(e_{\max}^{(k)}) - \log_{10}(\varepsilon)) \tag{27}$$

where B is an arbitrary integer number used to control the growth in the number of mesh intervals. Then, the curvature density function $\rho(\tau^{(k)})$

$$\rho(\tau^{(k)}) = c\kappa(\tau^{(k)})^{1/3} \tag{28}$$

is used to determine the locations of the new mesh points [21,69,65,66,70]. In Eq. (28), c is a constant chosen so that

$$\int_{-1}^{1} \rho(\zeta)d\zeta = 1 \tag{29}$$

and subsequently, the cumulative density function is defined as

$$F(\tau^{(k)}) = \int_{-1}^{\bar{t}} \rho(\zeta)d\zeta \tag{30}$$

Accordingly, the new mesh points are located so that

$$F(\tau_i^{(k)}) = \frac{i-1}{n_k}, \quad 1 \le i \le n_k + 1 \tag{31}$$

This latter relationship represents the fact that the density of the subinterval of the original mesh subinterval k is proportional to $F(\tau^{(k)})$ as the ith subinterval starts at $\tau_{i-1}^{(k)}$ and the $(i+1)$th subinterval starts at $F(\tau_i^{(k)}) - F(\tau_{i-1}^{(k)}) = n_k^{-1}$.

To summarize, compared with other PS methods, the hp-AR method relies on the low-degree approximation polynomials within each mesh subinterval resulting in a remarkably sparse configuration enabling the NLP solver to easily and efficiently compute the solution. In addition to that, due to the exponential convergence of the PS methods (using the orthogonal collocation points), in the mesh intervals where the solution is smooth, an accurate solution can be achieved using only a relatively small number of collocation points. As a result, the hp-AR algorithm assures simultaneous improvement in terms of solution accuracy and computational efficiency.

4. Evaluation of the hp-AR method for solving AUV docking HBVP

This section evaluates optimality and feasibility of solutions of the AUV underwater docking HBVP, obtained using the hp-AR method outlined in Section 3. Numerical analysis was conducted on a desktop computer with an Intel i7 3.40 GHz quad-core processor using the Interior Point OPTimizer (IPOPT) in the MATLAB® R2018a development environment. The accuracy tolerance of $\varepsilon = 10^{-6}$ was set for the discretized dynamic and path constraints. The initial guesses on states and controls were obtained using quasi-optimal solutions provided by the IDVD method as employed in [19,34,71–74] (as reported in [7,19,34], implementing IDVD assures convergence of PS methods and speeds it up).

4.1. Docking in a controlled operating environment

Let us start with a simple docking operation scenario involving no obstacles and assuming a full knowledge of the underwater currents field and DS position/orientation (Scenario 1). The AUV's initial state vector is defined as

$$\mathbf{X}_0 = [50\text{ m}, 50\text{ m}, 5\text{ m}, 10°, 0.3\text{ m s}^{-1}, 0\text{ m s}^{-1}, 0°\text{s}^{-1}]^T \tag{32}$$

The final state vector is defined by the DS position $\mathbf{P}_{DS} = [150\text{ m}, 75\text{ m}, 10\text{ m}]^T$ (Fig. 2) and heading $\psi_{DS} = 225°$ resulting in

$$\mathbf{X}_f = [150\text{ m}, 75\text{ m}, 10\text{ m}, 45°, 0.4\text{ m s}^{-1}, 0\text{ m s}^{-1}, 0°\text{s}^{-1}]^T \tag{33}$$

It is assumed that the AUV fits within the DS in such a way that once docked the AUV's centre of gravity coincides with the coordinates of the DS location; otherwise, some offset should be applied. A planar current is represented by

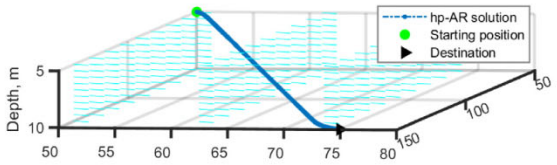

Fig. 3. 3D AUV path.

$$\mathbf{C} = [0.25\text{ m s}^{-1}, 0.25\text{ m s}^{-1}, 0]^T \tag{34}$$

The specific values for parameters imposing physical limitations on the AUV's states and controls, as defined by Eq. (9), are provided in [19]. For this scenario, the hp-AR method starts with two subintervals ($K = 2$) and four collocation points per interval ($N_k = 4$). The maximum number of mesh refinements is bounded by 10.

Figs. 3–7 illustrate the simulation results. Shown in Fig. 3, the AUV manoeuvres smoothly from the starting point and steers itself into the DS position. The manoeuvre concludes with the vehicle aligned with the DS entrance, i.e. featuring a 45° heading (Fig. 4(a)). Because of the expanded AUV model (8) used in the trajectory optimization the yaw rate becomes zero at the end, thus assuring smooth arrival (Fig. 4(b)). As shown in Fig. 5, the surge and heave components of the velocity vector increase rapidly at the commencement of the manoeuvre and then remain constant throughout the entire trajectory, decreasing only towards the end to meet the final conditions. This velocity vector profile is enabled by the corresponding control inputs depicted in Fig. 6, which shows that the controls remain constant throughout most of the trajectory, changing only towards the end to enforce the final conditions. It should be noted that the values of the control variables at commencement are not set to those corresponding to the state vector $\mathbf{X}_0$. This is due to the nature of the model (8), not allowing setting the boundary conditions on controls. Indeed, expanding the model to include the current control component rates can fix the problem, however adding this capability would drastically reduce the computational effectiveness of the algorithm rendering it difficult to compute the solution in real time on-board an AUV [75].

Fig. 7(a) shows the final count and distribution of the computational nodes for the hp-AR method and compares those to the solutions obtained using the LPS and CPS methods. To be consistent with [19], the number of nodes for the latter two methods is fixed and set to 50 (as shown in [34] this is a minimal number of nodes required to obtain a reasonable solution: with less nodes there is a lot of oscillations caused by numerical instability; having more nodes drastically increases the required computational time). The plot reveals that the hp-AR algorithm employs six times as many nodes as those of the LPS and CPS algorithms. Compared with other two methods, while the number of collocation points during midcourse is about the same, the hp-AR method features tenfold node concentration near the endpoints (that is where the majority of "extra" nodes are utilized). As a result, the Hamiltonian for the hp-AR solution resembles a constant as it is supposed to be for a time-independent system [75], while for LPS and CPS it slightly varies towards the end of trajectory (Fig. 7(b)) (suggesting that having even 50 nodes is probably not good enough). This means that the numerical solution obtained by the hp-AR method is "more optimal" (or more accurate) than the other two. The time histories for the components of the costate vector provided in Figs. 8–10 further

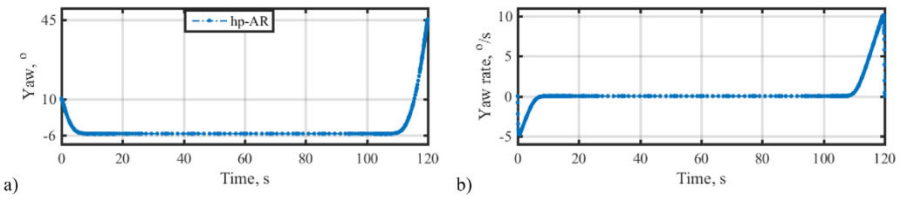

Fig. 4. Time histories of yaw (a) and yaw rate (b).

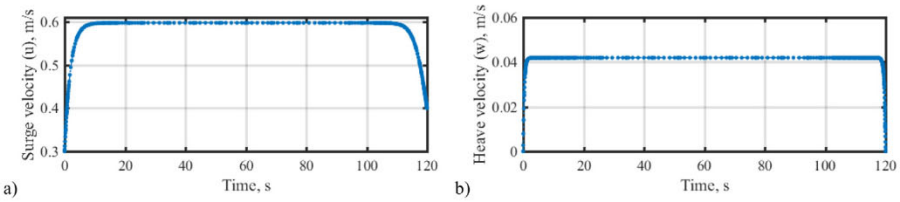

Fig. 5. Time histories of surge (a) and heave (b) velocity components.

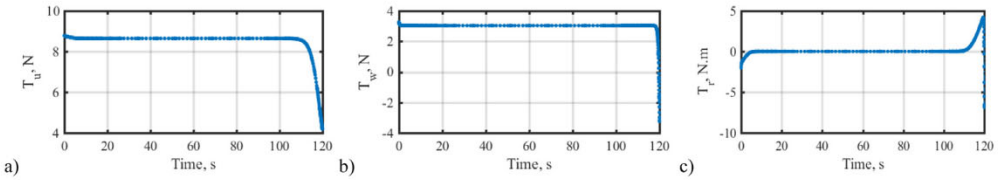

Fig. 6. Time histories of the T_u (a), T_w (b) and T_r (c) controls.

support this conclusion (in all figures the LPS solution resides on the top while the hp-AR method solution shows at the bottom).

Fig. 8 reveals chattering and oscillations of costates λ_x, λ_y, and λ_z, especially at endpoints for CPS and even more for LPS, while the hp-AR method-based solution provides a stable behaviour with good numerical precision in holding the coordinate costates constant [19]. The time histories for surge and heave costates for the LPS- and CPS-based solutions are consistent with the corresponding histories for the hp-AR method-based solution exhibiting a more accurate solution towards the end of trajectory (Fig. 9). The time-histories for costates λ_ψ and λ_r (Fig. 10) indicate the presence of a singular arch, when $\psi \equiv const$ and therefore $\dot{\psi} = r = \dot{r} = T_{r;opt} \equiv 0$ [19]. Again, the hp-AR method-based solution seems to be more accurate, especially for costate λ_r towards the end of the trajectory.

Table 1 presents a numerical comparison of respective performances of the hp-AR method with LPS and CPS counterparts based on several metrics related to solution optimality and computational efficiency. Specifically, these metrics include the value of the normalized cost function J (Eq. (10)), an estimate of the control input expenditure savings (ES) compared to the case of manoeuvring at control bounds, $T_u(t) = T_u^{\max}$, where

$$ES = 100(1 - \sqrt{J}), \tag{35}$$

the norm of the solution error (E) needed to meet the corresponding accuracy tolerance of 10^{-6}, the number of iterations during an optimization run (*Iter*), the number of cost function evaluations (*F-Count*), and the computational time required to obtain the optimal solution (t_{CPU}).

As observed from Table 1, all three PS methods deliver about the same value of the cost function J, while the hp-AR method assures the smallest error E. Although, the authors could not compare computational effectiveness of the three approaches explicitly (because different optimization engines were used), the convergence rate of the hp-AR method seems to be significantly

Table 1
Comparison of optimality and computational efficiency of solutions obtained with the LPS, CPS, and hp-AR methods.

Method	Metric					
	J	*ES*, %	*E*	*Iter*	*F-Count*	t_{CPU}, s
LPS	**0.2059**74	**54.61**56	7.8×10^{-7}	112	56,613	103.1
CPS	**0.2059**59	**54.61**73	5.7×10^{-7}	107	54,108	102.4
hp-AR	**0.2059**84	**54.61**45	2.7×10^{-7}	75	4,704	2.84[a]

[a]The hp-AR method utilizes a compiled IPOPT solver as opposed to the *fmincon* interpretative solver employed for the LPS and CPS methods.

better than that of the LPS and CPS. This statement is supported by more than an order of magnitude lower number of cost function evaluations *F-Count* and number of iterations required for the solution to converge to a given tolerance *Iter*.

The difference in *F-Count* can be explained as follows. Firstly, the differentiation matrix of the hp-AR method is rectangular (Eq. (19)) and has a full rank whilst those of the LPS and CPS methods are square and singular. As a result, the hp-AR method can be written equivalently in either differential or implicit integral form, whereas the LPS and CPS methods do not have an equivalent implicit integral form. Secondly, the hp-AR discrete costate transformation matrix has a full rank, while the LPS and CPS counterparts are rank-deficient. Consequently, the hp-AR costate approximations show exponential convergence while the LPS and CPS ones demonstrate potential non-converging solutions. The oscillatory behaviour observed in the LPS and CPS costates are due to the oscillatory nature of the null space of the LPS and CPS discrete costate transformation matrices [31].

To summarize, while all three methods produce about the same solutions (for Scenario 1), the superb computational effectiveness, accuracy and stability of the hp-AR method makes it a good candidate for on-line implementation.

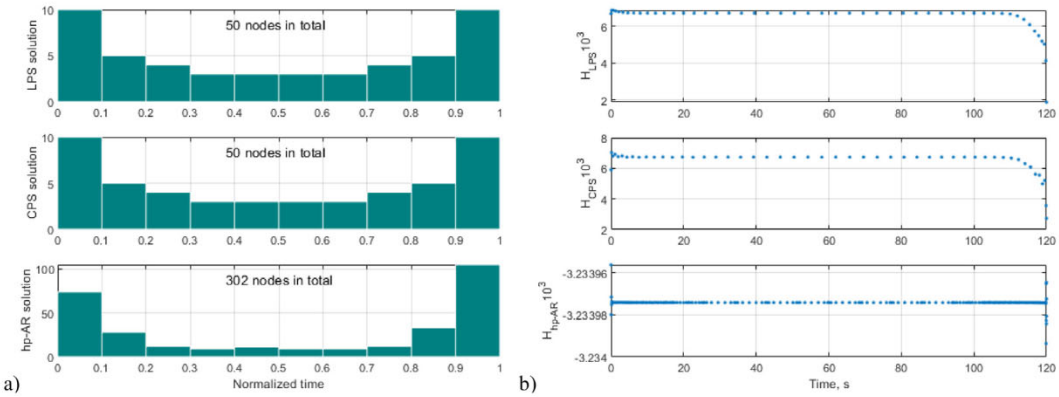

Fig. 7. Distribution of computational nodes (a), and Hamiltonian time histories (b) for the LPS, CPS and hp-AR methods.

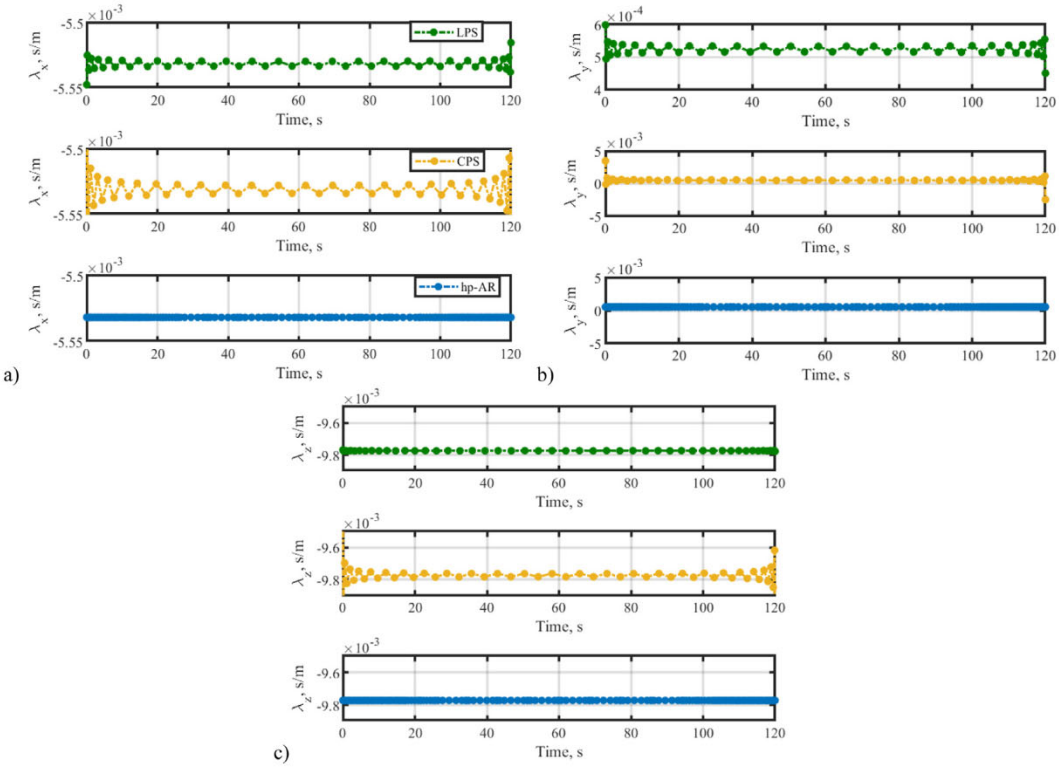

Fig. 8. Time histories of x- (a), y- (b), and z-coordinate (c) costates for the LPS, CPS and hp-AR methods (top to bottom).

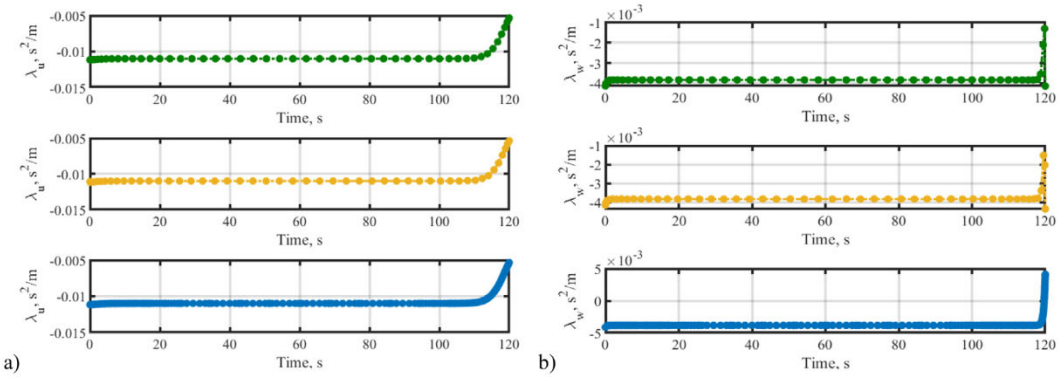

Fig. 9. Time histories of the surge (a) and heave (b) velocity costates.

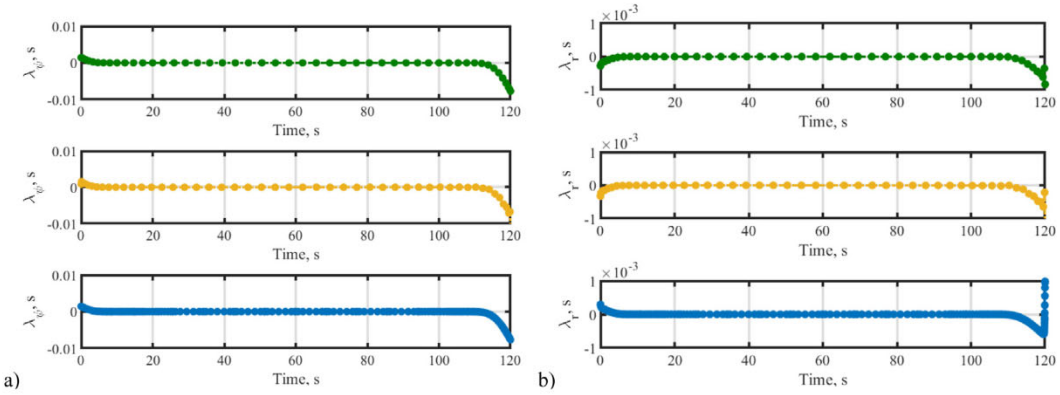

Fig. 10. Time histories of yaw (a) and yaw rate (b) costates.

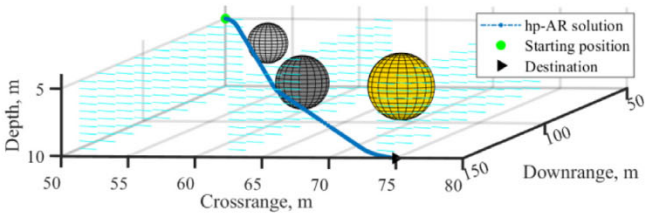

Fig. 11. 3D collision-free AUV path provided by the hp-AR (a) and LPS (b) methods in Scenario 2.

4.2. Docking guidance system performance in a cluttered field

To check whether conclusions of the previous section still hold, let us now consider a slightly more complex scenario where several no-fly zones (14) were added (Scenario 2)

$$NF_1: \begin{cases} \mathbf{P}_{NF} = [65, 55, 6]^T \text{ m} \\ r_{NF} = 1.5 \text{ m} \end{cases}, NF_2: \begin{cases} \mathbf{P}_{NF} = [101, 62, 7]^T \text{ m} \\ r_{NF} = 2 \text{ m} \end{cases},$$
$$NF_3: \begin{cases} \mathbf{P}_{NF} = [106, 70, 7]^T \text{ m} \\ r_{NF} = 2.5 \text{ m} \end{cases} \tag{36}$$

The parameters of the hp-AR method are exactly the same as for Scenario 1. Fig. 11 shows a collision-free trajectory generated by the hp-AR method in this case. This trajectory is smooth and satisfies all constraints including NF zone avoidance (Figs. 12–14).

Compared to Scenario 1, the number of nodes for the LPS and CPS methods was increased from 50 to 80. However, even with the increased number of nodes, both LPS and CPS methods converged to the same (local minimum) solution, which is different from that of the hp-AR method. Fig. 15 shows this solution going around the first two NF zones as opposed to just around the second one as in Fig. 11.

Similar to Figs. 7(a), 16(a) compares distributions of the collocation nodes for LPS, CPS, and hp-AR methods in Scenario 2. Overall, the hp-AR method solution features five times more nodes compared to the other two solutions, and again most of these are concentrated at locations along the trajectory where the AUV changes its course (that is where additional collocation points are introduced following the curvature-based adaptive procedure introduced in Section 3.2). Fig. 16(b) shows the time histories for the Hamiltonian of the system (compare with Fig. 7(b)). Yet again, the hp-AR method assures a more stable behaviour satisfying the underlying theory of Hamiltonian being constant (while both the LPS and CPS methods struggle to converge to a solution even with the increased number of nodes).

In the presence of path constraints, in addition to maintaining all necessary conditions, the tangential velocity vector condition associated with the path constraint (14) must be held [75]. This implies that in the regions where the path constraint is active, a jump or discontinuity in a costate time history (numerical solution) might be observed. This happens the case when a trajectory goes going around a NF zone (Fig. 17). Again, the hp-AR method solution seems to be more feasible (accurate) identifying the

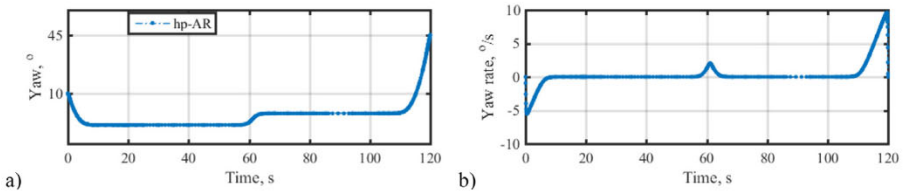

Fig. 12. Time histories of yaw (a) and yaw rate (b) provided by the hp-AR method in Scenario 2.

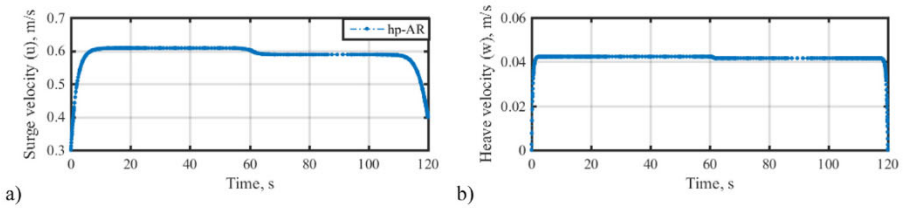

Fig. 13. Time histories of surge (a) and heave (b) velocities provided by the hp-AR method in Scenario 2.

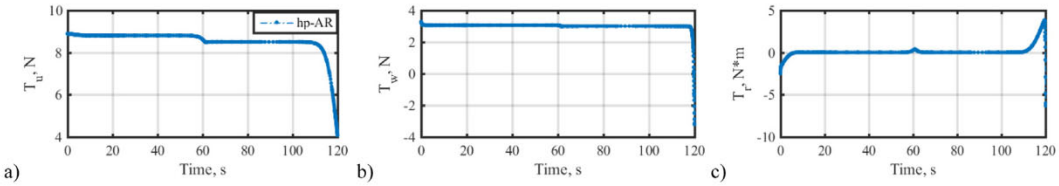

Fig. 14. Time histories of T_u (a), T_w (b) and T_r (c) controls provided by the hp-AR method in Scenario 2.

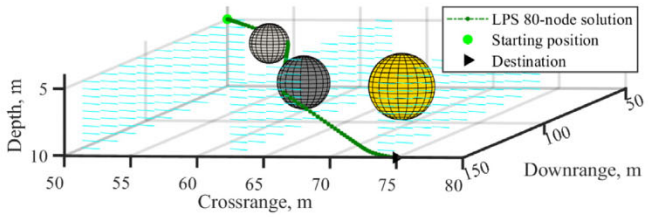

Fig. 15. 3D collision-free AUV path provided by LPS (and CPS) method in Scenario 2 (compare with Fig. 11).

moment when trajectory touches a second NF zone precisely, while the LPS and CPS methods both converge to a different solution.

Similar to Scenario 1, the hp-AR method solution exhibits no costate oscillations at the end of trajectory (Figs. 18 and 19). Time histories for costates λ_ψ and λ_r suggest the existence of a singular arc control yet again. While there are two switching points for the LPS and CPS solutions, there is only one for the hp-AR method solution (Fig. 19).

Similar to Tables 1, 2 provides a quantitative comparison of solution optimality and computational efficiency for the minimum-energy docking operation in the cluttered environment (Scenario 2). As mentioned earlier, this time, the hp-AR solution (touching a single NF zone) differs from other two (touching two NF zones). As a result, the cost function value for LPS and CPS solutions is about 2% worse than that of the hp-AR method solution. Even though t_{CPU} has increased (more so for the LPS and CPS methods than for the hp-AR method), all conclusions derived from analysing the results of Scenario 1 simulation presented in Section 4.1, seem to hold for Scenario 2 simulation as well.

5. Assessment of the hp-AR method solutions using SITLSP

This section evaluates the numerical solution produced by the three different PS methods from the standpoint of their applicability within an actual real-time control architecture of a high-fidelity SITLSP incorporating realistically modelled components of the AUV and emulating the vehicle's real behaviour while operating in an undersea environment [19,20]. It first addresses the issue of trajectory trackability and then proceeds with exploring the robustness of solutions provided by PS methods.

5.1. Trackability assessment

Consider Scenario 1 again (Section 4.1). This time however, the controls computed using the hp-AR method are interpolated over the even intervals to be used within SITLSP. The trackability of these trajectories (using a feed forward controller) is assessed in a numerical sense using the set of metrics computed at the final point (docking)

$$\Delta\eta = \sqrt{(x_f^{SITLSP} - x_{DS})^2 + (y_f^{SITLSP} - y_{DS})^2 + (z_f^{SITLSP} - z_{DS})^2} \le h_{DS} \tag{37}$$

Table 2
Solution optimality and computational efficiency of the LPS and CPS methods compared to the hp-AR method.

	Metric		
Method	J	ES, %	t_{CPU}, s
LPS	0.2113	54.03	516.2
CPS	0.2111	54.05	499.3
hp-AR	0.2067	54.53	8.94[a]

[a]The hp-AR solution utilizes a compiled IPOPT solver as opposed to the *fmincon* interpretative solver.

Table 3
Comparison of SITLSP trackability performance for the hp-AR with LPS and CPS methods for Scenario 1.

	Metric				
Method	$\Delta\eta$, m	$\Delta\psi$, °	Δu, m/s	Δw, m/s	Δr, °/s
LPS	0.21	8.2	9.9×10^{-5}	2.5×10^{-4}	6.0
CPS	0.21	8.4	1.0×10^{-4}	2.9×10^{-4}	6.0
hp-AR	0.20	7.7	9.2×10^{-5}	6.7×10^{-5}	5.9

$$\Delta\psi = \left|\psi_f^{SITLSP} - \psi_{DS}\right| \le 0.15\alpha_{DS} \tag{38}$$

$$\Delta u = \left|u_f^{SITLSP} - u_f\right| \le \varepsilon_u \tag{39}$$

$$\Delta w = \left|w_f^{SITLSP} - w_f\right| \le \varepsilon_w \tag{40}$$

$$\Delta r = \left|r_f^{SITLSP} - r_f\right| \le \varepsilon_r \tag{41}$$

The superscript SITLSP used in Eqs. (37)–(41) indicates the final state values obtained after the SITLSP run, while the subscript f indicates the desired final states (where AUV would be in the case of precise tracking of the reference controls). The discrepancies $\Delta\eta$ and $\Delta\psi$ correspond to the final position and direction errors, respectively, indicating to what extent the generated controls enable the achievement of the final docking approach (see notations of Fig. 2). Additionally, Δu, Δw, and Δr represent the terminal errors on surge, heave and yaw rate, respectively. These errors should satisfy tolerances ε_i, $i \in \{u, w, r\}$ that are set to 10^{-4}m/s for surge and heave terminal errors and 6°/s for the terminal yaw rate error.

Figs. 20–22 show the results of execution of a minimum-energy solution (generated by the hp-AR method for Scenario 1) on SITLSP. Overall, the reference controls tracking performance is acceptable even though SITLSP uses a higher-fidelity model compared to that used for trajectory optimization. This implicitly attests that the lower-fidelity model (4), (5), (7), (8) still captures the essence of AUV dynamics.

As seen, the SITLSP outputs in the beginning of the run differ from the (reference) ones produced by the hp-AR method. This is due to the fact that optimal solution in formulation (4), (5), (7), (8) cannot possibly satisfy the controls' boundary conditions or in other words do not start from a stable configuration, i.e., Eq. (8) does not hold. As a result, it takes some time for the actual AUV dynamics to catch up with the reference trajectory produced by the hp-AR method. Nevertheless, none of the SITLSP outputs violate the constraints (20) thus verifying the trackability of the trajectories generated by the hp-AR method and therefore its applicability for implementation on an AUV's onboard computer.

Table 3 compares the SITLSP tracking metrics of the hp-AR method in Scenario 1 with those of LPS and CPS counterparts. In all cases, tolerances (37)–(41) are easily met when using any of the three PS methods, although the hp-AR method exhibits a slight advantage compared to the other two methods. Similar results are demonstrated in Table 4 for Scenario 2 (Section 4.2).

5.2. Robustness assessment

A series of Monte Carlo (MC) simulations varying boundary conditions was conducted to assess the robustness of the hp-AR

Table 4
Comparison of SITLSP trackability performance for the hp-AR with LPS and CPS methods for Scenario 2.

Method	Metric				
	$\Delta\eta$, m	$\Delta\psi$, °	Δu, m/s	Δw, m/s	Δr, °/s
LPS	0.39	6.0	6.2×10^{-5}	2.5×10^{-5}	5.2
CPS	0.38	6.1	6.4×10^{-5}	3.0×10^{-6}	5.2
hp-AR	0.19	7.4	8.4×10^{-5}	8.6×10^{-5}	5.7

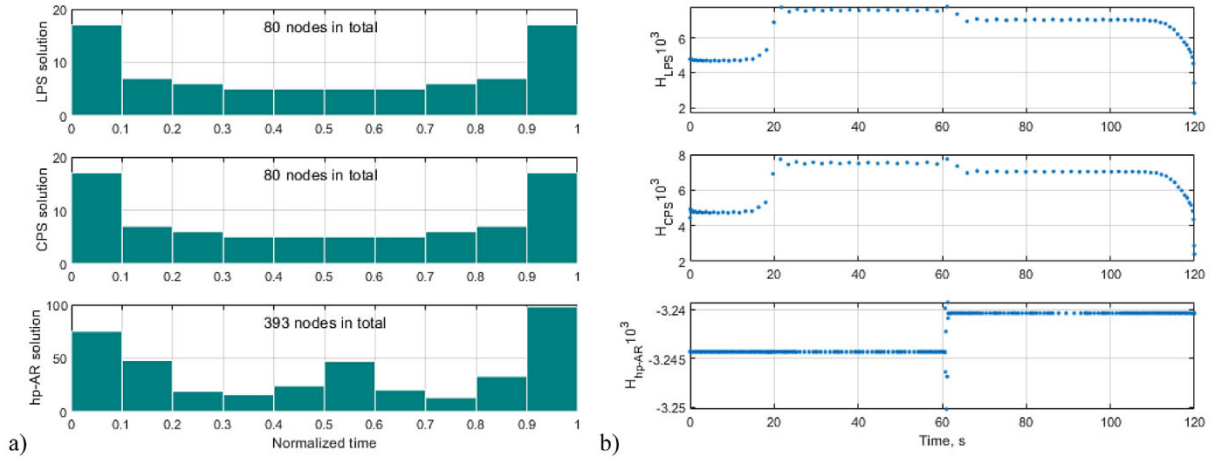

Fig. 16. Distribution of computational nodes (a), and Hamiltonian time histories provided by the hp-AR, LPS, and CPS methods in Scenario 2.

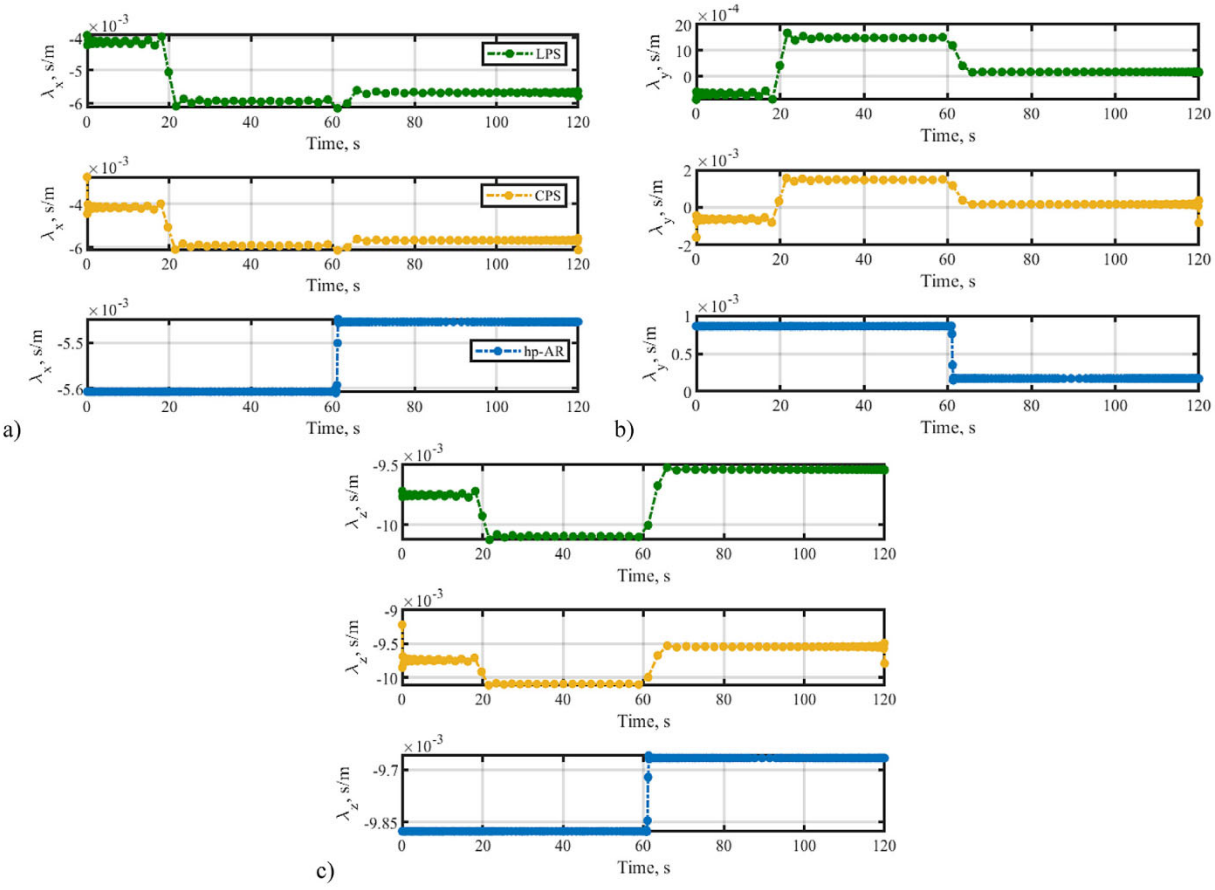

Fig. 17. Time histories of x- (a), y- (b), and z-coordinate (c) costates provided by the hp-AR, LPS, and CPS methods in Scenario 2.

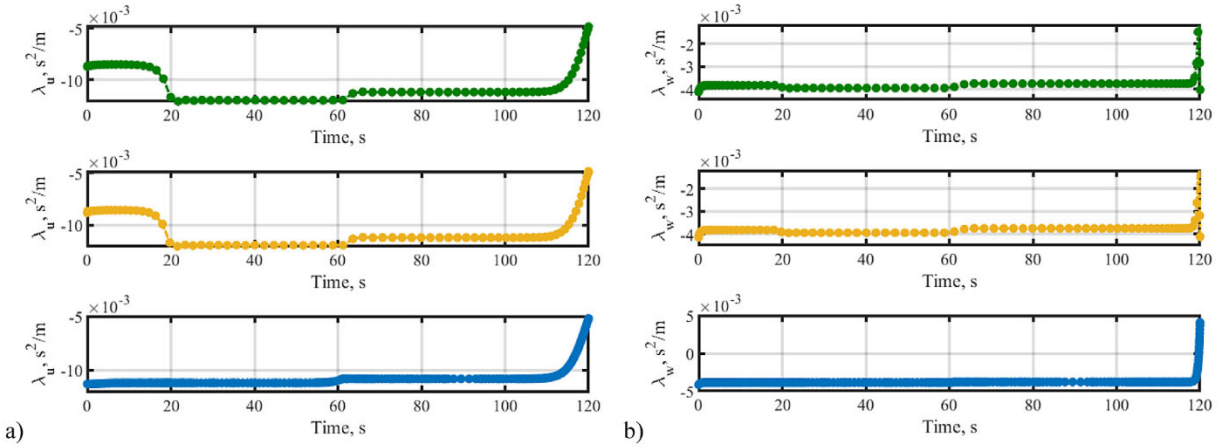

Fig. 18. Time histories of surge (a) and heave (b) velocity costates provided by the hp-AR, LPS, and CPS methods in Scenario 2.

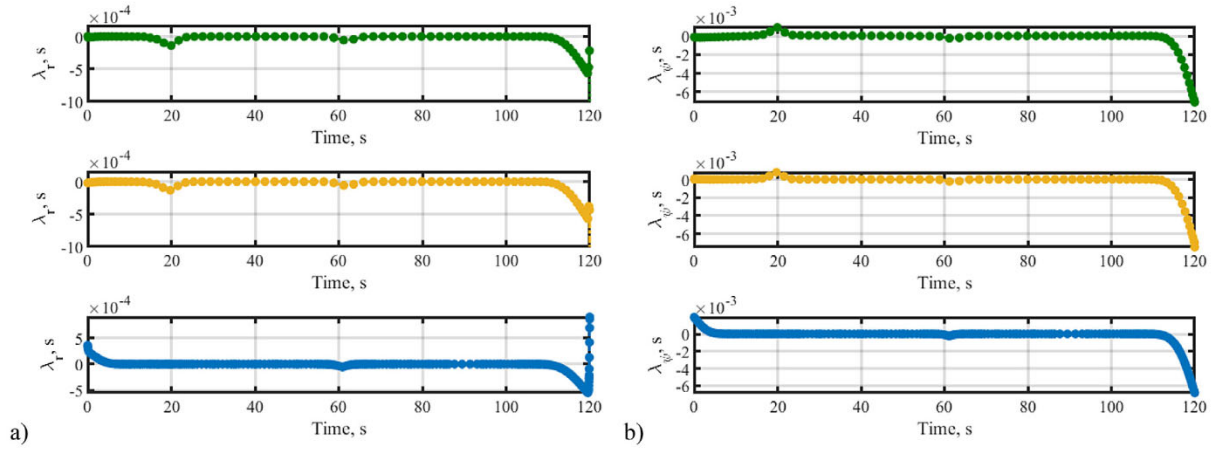

Fig. 19. Time histories of yaw (a) and yaw rate (b) costates provided by the hp-AR, LPS, and CPS methods in Scenario 2.

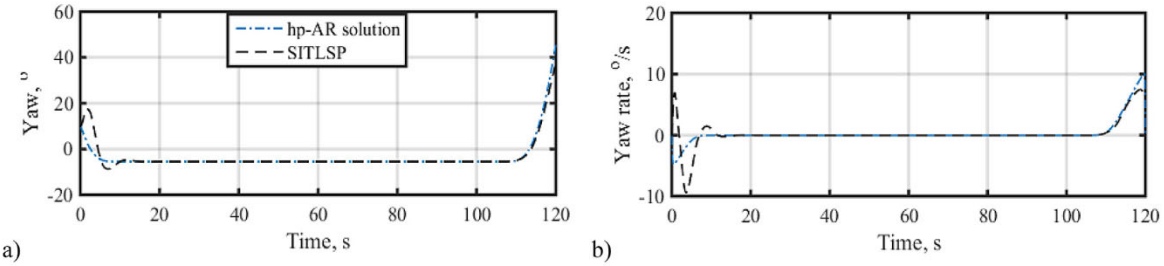

Fig. 20. Realization of yaw (a) and yaw rate (b) time histories on SITLSP.

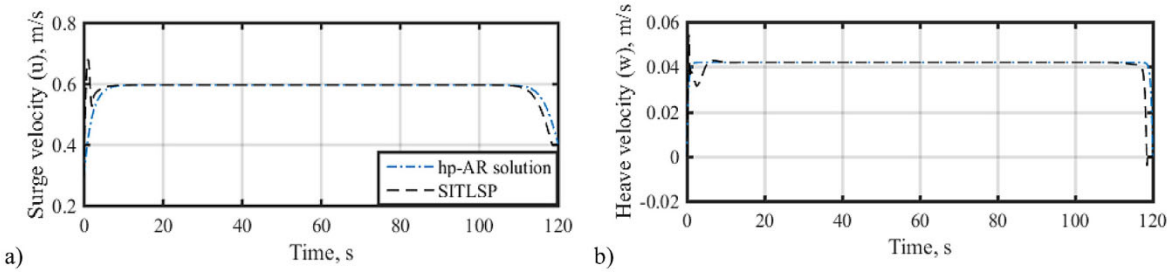

Fig. 21. Realization of surge (a) and heave (b) velocities on SITLSP.

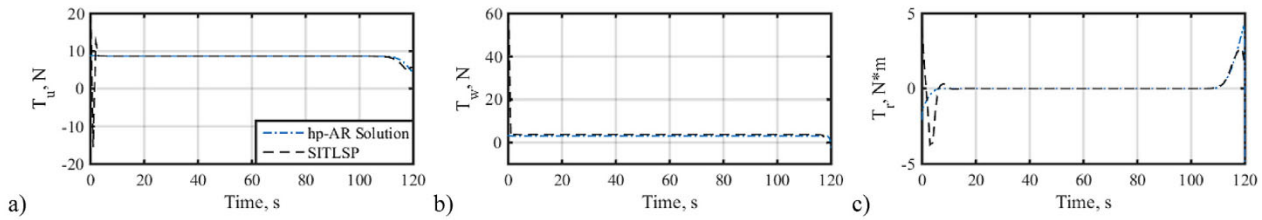

Fig. 22. Realization of T_u (a), T_w (b) and T_r (c) controls on SITLSP.

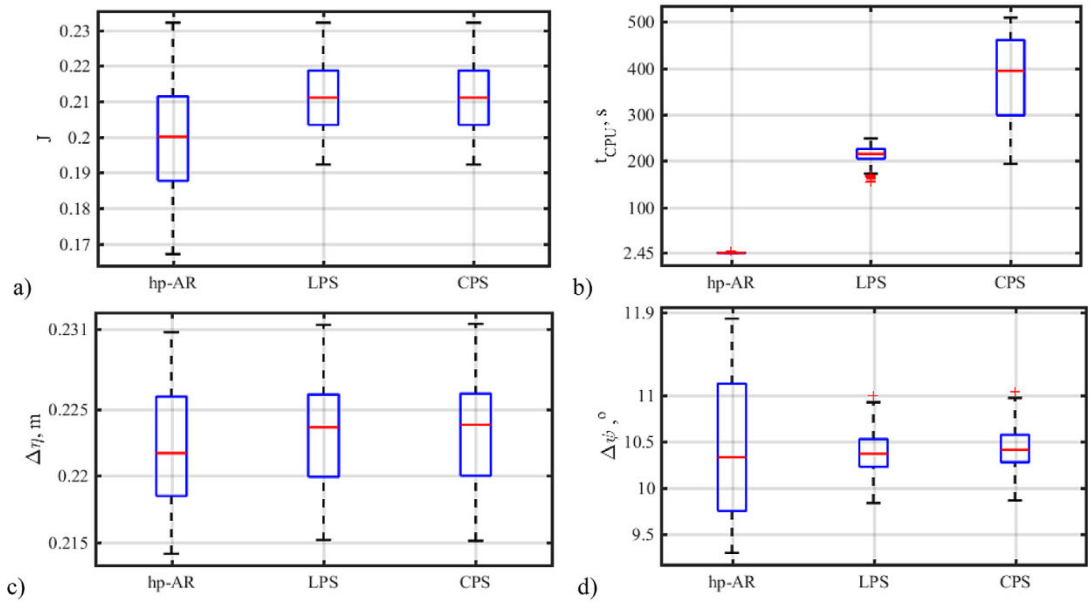

Fig. 23. Robustness assessment of the hp-AR against LPS and CPS in MC simulations with initial pose variations.

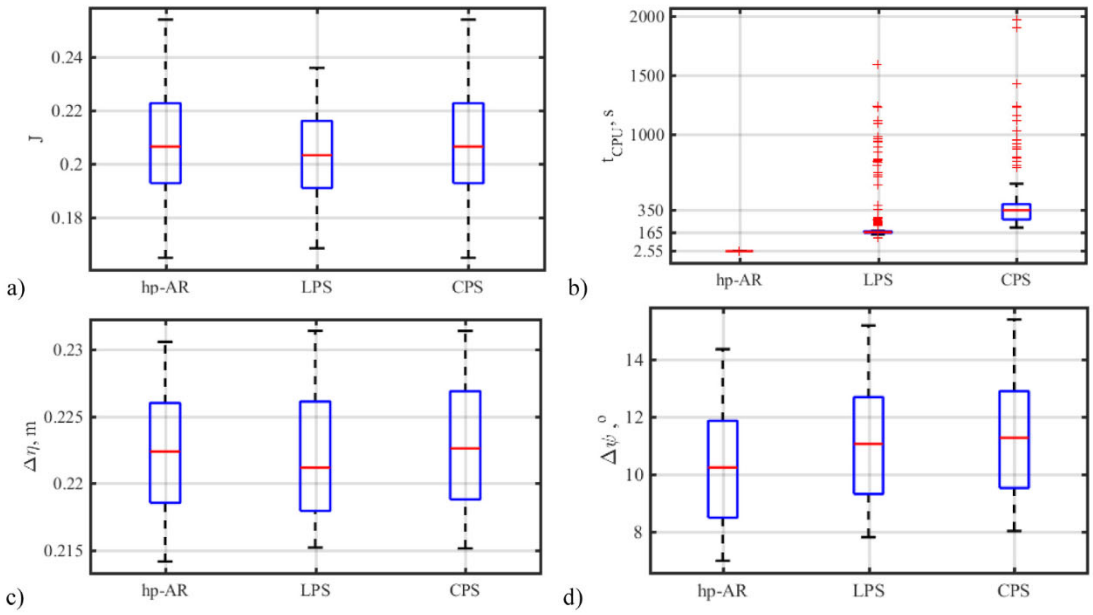

Fig. 24. Robustness assessment of the hp-AR against LPS and CPS in MC simulations with DS pose variations.

method and determine the effectiveness of the SITLSP controller in tracking the optimal trajectory. In order to emulate any inaccuracies that may occur in modelling the AUV's hydrodynamics, the corresponding hydrodynamic coefficients used in SITLSP were also varied by as much as by 10% compared to those of Eq. (8).

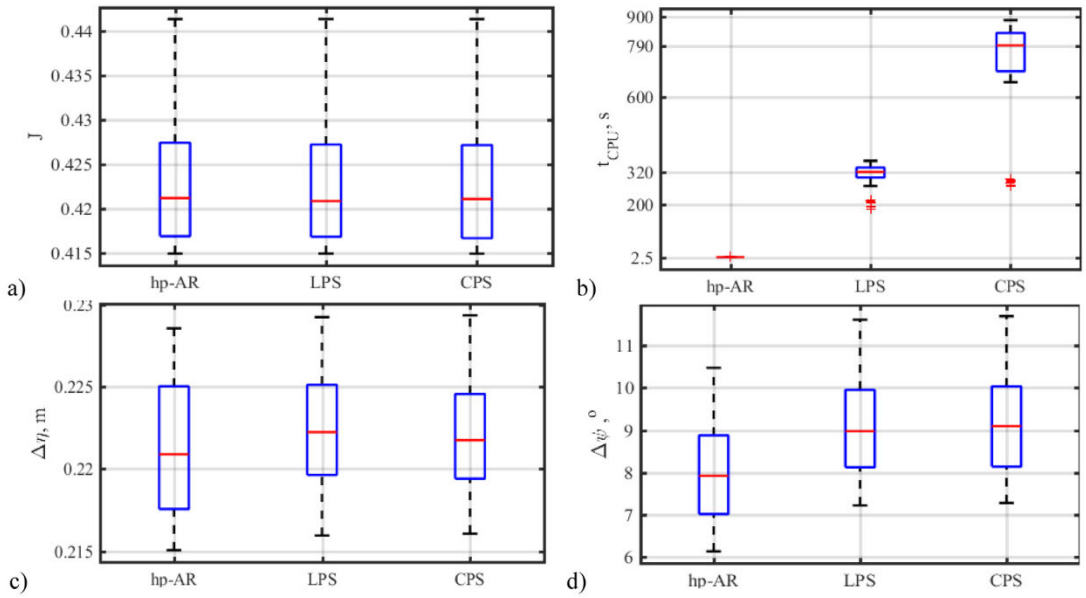

Fig. 25. Robustness assessment of the hp-AR against LPS and CPS in MC simulations with the current variations.

The first set of MC trials involved a set of 200 runs with a variable initial state vector

$$\mathbf{X}_0 = [(50 + 5\rho_1)\ \text{m}, (50 + 5\rho_2)\ \text{m}, (5 + 2\rho_3)\ \text{m}, (10 + 20\rho_4)^\circ, \\ 0.3\ \text{m s}^{-1}, 0\ \text{m s}^{-1}, 0^\circ\text{s}^{-1}]^T, \tag{42}$$

where the parameters $\rho_i, i = 1, 2, 3, 4$ were uniformly distributed over the range $[-1; 1]$. All other conditions, aside from the hydrodynamic coefficients, were kept the same as in Scenario 1 of Section 4.1. Fig. 23 shows the results of MC simulations for the hp-AR method and compares these with the performance of LPS and CPS methods.

The results reveal that for all MC runs, all three PS methods were able to converge to a solution. The accuracy achieved using the hp-AR method was superior to the accuracies achieved using the other two methods (Fig. 23(a)). While the computation time for the hp-AR method was consistent for all variations, the performances of other two methods depended significantly on the particular set of initial conditions (Fig. 23(b)). The reason was due to the difficulty faced by the LPS and CPS methods in finding the singular-arch switching instances with just 50 nodes, whereas the hp-AR method simply added as many additional nodes as necessary, or increased the order of the approximating polynomial (as discussed in Section 3.2). In terms of precision (satisfying the final pose), all three methods exhibited about the same performance (Fig. 23(c), (d)). It should be noted that the errors shown in Fig. 23(c), (d) are mostly attributable to a 2-second lag in the sliding mode controller of SITLSP as mentioned in [19].

The next set of 200 MC trials was executed with varied final boundary conditions

$$\mathbf{X}_f = [(150 + 5\rho_1)\ \text{m}, (75 + 5\rho_2)\ \text{m}, (10 + 2\rho_3)\ \text{m}, \\ (45 + 20\rho_4)^\circ, 0.4\ \text{m s}^{-1}, 0, 0]^T, \tag{43}$$

with ρ_i, $i = 1, 2, 3, 4$ uniformly distributed over the range $[-1; 1]$ along with a 10% uncertainty applied to all hydrodynamic coefficients used in the SITLSP. The results are shown in Fig. 23.

In this case, in contrast to the LPS and CPS methods, the hp-AR method exhibited the very similar results to those shown in Fig. 23. As seen from Fig. 24(b), on many occasions both LPS and CPS had difficulties converging to the correct solution, even though all three methods were initialized exactly the same using the IDVD-method provided solution. The hp-AR method also proved to be slightly more robust with respect to satisfying the final heading constraint (38). In this series of 200 MC runs, the hp-AR method only had 12 violations, whereas the LPS and CPS methods have 31 and 37 violations, respectively.

The third set of the MC tests considered an impact of uncertainty in measuring the current. To this end, the current components used to produce a solution in Scenario 1 (Section 4.1), were varied within SITLSP as

$$\mathbf{C} = [0\ \text{m s}^{-1}, (0.3 + 0.1\rho)\ \text{m s}^{-1}, 0]^T \tag{44}$$

with ρ being uniformly distributed over the range $[-1; 1]$. Fig. 25 depicts the results of the MC simulations in this case. The results reveal that all 200 MC simulations converged to bout the same value of J (Fig. 25(a)) with the hp-AR method demonstrating a better result in terms of computational efficiency (Fig. 25(b)).

Compared to the previous cases featuring no currents, it now took approximately twice as much energy expenditure per second to overcome the current disturbance for an acceptable docking, no matter which PS was utilized (Fig. 25(a)). By tracking the produced optimal trajectory on SITLSP, the hp-AR method seems to assure slightly better performance (Fig. 25(c), (d)).

6. Conclusion

In this paper, it was shown that the hp-adaptive Radau method can effectively be used to solve a two-point boundary-value optimal control problem for AUV executing a short-range short-duration energy-conservative underwater docking mission. Despite utilizing a high-fidelity AUV model that encapsulates realistic vehicular and environmental constraints such as the saturation levels of the AUV's thrusters, ocean current disturbances, and presence of multiple no-fly zones, the hp-adaptive Radau method had no issues with generating a numerically stable, trackable optimal solution. This is in a clear contrast with other non-adaptive pseudospectral methods based on Lagrange and Chebyshev global polynomial approximations. Even though the solution features more computational nodes, when used with the Inverse Dynamics in the Virtual Domain method to generate an initial guess for the follow-up optimization, the computational effectiveness of the hp-adaptive Radau method is several orders of magnitude

better than that of other pseudospectral methods. The effectiveness, fidelity, and robustness of the hp-AR method-based reference trajectory generator were verified via extensive testing on a realistic software-in-the-loop simulation platform for a prototype of AUV. Monte Carlo simulations included varying initial and terminal conditions, ocean currents and emulating uncertainties in the values of AUV parameters. The computational effectiveness, accuracy and robustness of the hp-adaptive (multi-interval) Radau method makes it a good candidate to be implemented and further tested on a real AUV, which is currently under way.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C.C. Eriksen, T.J. Osse, R.D. Light, T. Wen, T.W. Lehman, P.L. Sabin, J.W. Ballard, A.M. Chiodi, Seaglider: A long-range autonomous underwater vehicle for oceanographic research, IEEE J. Ocean. Eng. 26 (2001) 424–436.

[2] S. Kemna, M.J. Hamilton, D.T. Hughes, K.D. LePage, Adaptive autonomous underwater vehicles for littoral surveillance, Intell. Serv. Robot. 4 (2011) 245–258.

[3] M. Grasmueck, G. Eberli, D. Viggiano, T. Correa, G. Rathwell, J. Luo, Autonomous underwater vehicle (AUV) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the Straits of Florida, Geophys. Res. Lett. 33 (23) (2006) 1–6.

[4] S.M. Zadeh, D.M. Powers, K. Sammut, A. Atyabi, A.M. Yazdani, A hierarchal planning framework for AUV mission management in a spatiotemporal varying ocean, Comput. Electr. Eng. 67 (2018) 741–760.

[5] M.D. Feezor, F.Y. Sorrell, P.R. Blankinship, J.G. Bellingham, Autonomous underwater vehicle homing/docking via electromagnetic guidance, IEEE J. Ocean. Eng. 26 (2001) 515–521.

[6] R.R. Stokey, B. Allen, T. Austin, R. Goldsborough, N. Forrester, M. Purcell, C. von Alt, Enabling technologies for REMUS docking: an integral component of an autonomous ocean-sampling network, IEEE J. Ocean. Eng. 26 (2001) 487–497.

[7] A.M. Yazdani, K. Sammut, O.Yakimenko, A. Lammas, A survey of underwater docking guidance systems, Robot. Auton. Syst. 124 (2020) 103382–103403.

[8] P.A. Wilson, Autonomous homing and docking tasks for an underwater vehicle, IFAC Proc. Vol. 42 (2009) 304–309.

[9] J.Y. Park, B.H. Jun, P.M. Lee, Y.K. Lim, J.H. Oh, Docking problem and guidance laws considering drift for an underactuated AUV, Oceans (2011) 1–7.

[10] W. Naeem, R. Sutton, S. Ahmad, Pure pursuit guidance and model predictive control of an autonomous underwater vehicle for cable/pipeline tracking, Proc. Inst. Mar. Eng. Sci. Technol. Part C (2004) 25–35.

[11] I. Spangelo, O. Egeland, Trajectory planning and collision avoidance for underwater vehicles using optimal control, IEEE J. Ocean. Eng. 19 (4) (1994) 502–511.

[12] M. Chyba, T. Haberkorn, S.B. Singh, R.N. Smith, S.K. Choi, Increasing underwater vehicle autonomy by reducing energy consumption, Ocean Eng. 36 (2009) 62–73.

[13] M.B. Loc, H.-S. Choi, S.-S. You, T.N. Huy, Time optimal trajectory design for unmanned underwater vehicle, Ocean Eng. 89 (2014) 69–81.

[14] R.P. Kumar, A. Dasgupta, C. Kumar, Real-time optimal motion planning for autonomous underwater vehicles, Ocean Eng. 32 (2005) 1431–1447.

[15] M. Chyba, T. Haberkorn, R.N. Smith, S.K. Choi, Design and implementation of time efficient trajectories for autonomous underwater vehicles, Ocean Eng. 35 (2008) 63–76.

[16] K. Kim, T. Ura, Optimal and quasi-optimal navigations of an AUV in current, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.

[17] K. Kim, T. Ura, Optimal guidance for autonomous underwater vehicle navigation within undersea areas of current disturbance, Adv. Robot. 23 (2009) 601–628.

[18] A.V. Rao, Trajectory optimization: a survey, in: Optimization and Optimal Control in Automotive Systems, Springer, 2014, pp. 3–21, ed.

[19] A.M. Yazdani, K. Sammut, O.A. Yakimenko, A. Lammas, Y. Tang, S.M. Zadeh, IDVD-based trajectory generator for autonomous underwater docking operations, Robot. Auton. Syst. 92 (2017) 12–29.

[20] A. Lammas, K. Sammut, F. He, 6-DoF navigation systems for autonomous underwater vehicles, in: A. Barrera (Ed.), Mobile Robots Navigation, InTech, 2010.

[21] P. Han, J. Shan, X. Meng, Re-entry trajectory optimization using an hp-adaptive Radau pseudospectral method, Proc. Inst. Mech. Eng. G (2012) 1–15.

[22] G.T. Huntington, A.V. Rao, Optimal reconfiguration of spacecraft formations using the Gauss pseudospectral method, J. Guid. Control Dyn. 31 (2008) 689–698.

[23] I.M. Ross, M. Karpenko, A review of pseudospectral optimal control: from theory to flight, Annu. Rev. Control 36 (2012) 182–197.

[24] B. Wu, D. Wang, E.K. Poh, G. Xu, Nonlinear optimization of low-thrust trajectory for satellite formation: Legendre pseudospectral approach, J. Guid. Control Dyn. 32 (2009) 1371–1381.

[25] J. Li, Fuel-optimal low-thrust formation reconfiguration via Radau pseudospectral method, Adv. Space Res. 58 (2016) 1–16.

[26] K.A. Baumgartner, S. Ferrari, A.V. Rao, Optimal control of an underwater sensor network for cooperative target tracking, IEEE J. Ocean. Eng. 34 (2009) 678–697.

[27] C. Francolin, A. Rao, C. Duarte, G. Martel, Optimal control of an autonomous surface vehicle to improve connectivity in an underwater vehicle network, J. Aerosp. Comput. Inf. Commun. 9 (2012) 1–13.

[28] A. Adhami-Mirhosseini, A.P. Aguiar, M.J. Yazdanpanah, Seabed tracking of an autonomous underwater vehicle with nonlinear output regulation, in: IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 3928–3933.

[29] Y. Zhuang, S. Sharma, B. Subudhi, H. Huang, J. Wan, Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm, Ocean Eng. 127 (2016) 190–199.

[30] G.T. Huntington, D. Benson, A.V. Rao, A comparison of accuracy and computational efficiency of three pseudospectral methods, in: AIAA Guidance, Navigation, and Control Conference, 2007, pp. 840–864.

[31] D. Garg, M. Patterson, W.W. Hager, A.V. Rao, D.A. Benson, G.T. Huntington, A unified framework for the numerical solution of optimal control problems using pseudospectral methods, Automatica 46 (2010) 1843–1851.

[32] D. Garg, M.A. Patterson, C. Francolin, C.L. Darby, G.T. Huntington, W.W. Hager, A.V. Rao, Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method, Comput. Optim. Appl. 49 (2011) 335–358.

[33] D. Garg, W.W. Hager, A.V. Rao, Pseudospectral methods for solving infinite-horizon optimal control problems, Automatica 47 (2011) 829–837.

[34] O. Yakimenko, Y. Xu, G. Basset, Computing short-time aircraft maneuvers using direct methods, J. Comput. Syst. Sci. Int. 49 (3) (2010) 145–176.

[35] J. Ventura, M. Romano, U. Walter, Performance evaluation of the inverse dynamics method for optimal spacecraft reorientation, Acta Astronaut. 110 (2015) 266–278.

[36] J.-Y. Park, Docking Problem and Guidance Laws Considering Drift for an Underactuated Autonomous AUV (Ph.D. dissertation), Korea Advanced Institute of Science and Technology, 2011.

[37] M. Abkowitz, Stability and Motion Control of Ocean Vehicles, MIT Press, 1969.

[38] M. Gertler, G. Hagen, Standard Equations of Motion for Submarine Simulation, Technical Report, Naval Ship Research and Development Center, 1967.

[39] T.I. Fossen, Guidance and Control of Ocean Vehicles, John Wiley & Sons, 1994.

[40] T.I. Fossen, Handbook of Marine Craft Hydrodynamics and Motion Control, John Wiley & Sons, ISBN: 978-1-119-99149-6, 2011.

[41] A.J. Healey, D. Lienard, Multivariable sliding model control for autonomous diving and steering of unmanned underwater vehicles, IEEE J. Ocean. Eng. 18 (3) (1993) 327–339.

[42] T.I. Fossen, Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles, Mar. Cybern. (2002) ISBN: 82-92356-00-2.

[43] M. Barisic, Z. Vukic, N. Miskovic, Formation guidance of AUVs using decentralized control functions, in: N.A. Cruz (Ed.), Autonomous Underwater Vehicles, InTech, ISBN: 978-953-307-432-0, 2011.

[44] T. Prestero, Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle (MS thesis), Massachusetts Institute of Technology, 2001.

[45] P. Ridley, J. Fontan, P. Corke, Submarine dynamic modeling, in: Australasian Conference on Robotics and Automation, 2003.

[46] E. Lind, M. Meijer, Simulation and Control of Submarines (MS thesis), Lund University, 2014.

[47] O.A. Yakimenko, S.P. Kragelund, Real-time optimal guidance and obstacle avoidance for UMVs, in: Autonomous Underwater Vehicles, InTech, 2011.

[48] D. Li, Y. Chen, J. Shi, C. Yang, Autonomous underwater vehicle docking system for cabled ocean observatory network, Ocean Eng. 109 (2015) 127–134.

[49] P. Ridao, J. Batlle, M. Carreras, Model identification of a low-speed UUV, IFAC Proc. Vol. 34 (7) (2001) 395–400.
[50] M.S. Triantafyllou, F.S. Hover, Maneuvering and control of marine vehicles, Course notes, Massachusetts Institute of Technology, 2002.
[51] L.A. Gonzales, Design, Modelling and Control of an Autonomous Underwater Vehicle (BE thesis), University of Western Australia, 2004.
[52] W. Wang, C.M. Clark, Modeling and simulation of the VideoRay Pro III underwater vehicle, Oceans (2006).
[53] G. Antonelli, Underwater Robots Motion and Force Control of Vehicle-Manipulator Systems, second ed., Springer, 2006, ISBN-13 978-3-540-31752-4.
[54] N. Mišković, Z. Vukić, M. Barišić, Identification of underwater vehicles for the purpose of autopilot tuning, in: A.V. Inzartsev (Ed.), Underwater Vehicles, InTech, ISBN: 978-953-7619-49-7, 2009.
[55] M.S. Arslan, N. Fukushima, I. Hagiwara, Optimal control of underactuated underwater vehicles with single actuator, in: A.V. Inzartsev (Ed.), Underwater Vehicles, InTech, ISBN: 978-953-7619-49-7, 2009.
[56] J.H.A.M. Vervoort, Modeling and Control of an Unmanned Underwater Vehicle (Ph.D. thesis), University of Canterbury, 2009.
[57] Y. Liu, R. Bucknall, Modelling and control of autonomous marine vehicles, in: S. Sharma, B. Subudhi (Eds.), Navigation and Control of Autonomous Marine Vehicles, in: IET Transportation Series, ISBN: 978-1-78561-338-8, 2019.
[58] F. Fanelli, Development and Testing of Navigation Algorithms for Autonomous Underwater Vehicles (Springer theses), ISBN: 978-3-030-15595-7, 2020.
[59] D. Perrault, N. Bose, S. O'Young, D. Williams, Sensitivity of AUV added mass coefficients to variations in hull and control plane geometry, Ocean Eng. 30 (5) (2003) 645–671.
[60] G.D. Watt, Modelling and Simulating Unsteady Six Degrees-of-Freedom Submarine Rising Maneuvers, Technical Report TR 2007-008, Defence Research and Development Canada, 2007.
[61] J.S. Geisbert, Hydrodynamic Modeling for Autonomous Underwater Vehicles using Computational and Semi-Empirical Methods (MS thesis), Virginia Polytechnic Institute and State University, 2007.
[62] J. Severholt, Generic 6-DOF Added Mass Formulation for Arbitrary Underwater Vehicles Based on Existing Semi-Empirical Methods (MS thesis), Royal Institute of Technology, 2017.
[63] M. Ataei, A. Yousefi-Koma, Development of a three-dimensional guidance system for long-range maneuvering of a miniature autonomous underwater vehicle, China Ocean Eng. 28 (6) (2014) 843–856.
[64] M. Ataei, Three-dimensional optimal path planning for waypoint guidance of an autonomous underwater vehicle, Robot. Auton. Syst. 67 (2015) 23–32.
[65] C.L. Darby, W.W. Hager, A.V. Rao, A preliminary analysis of a variable-order approach to solving optimal control problems using pseudospectral methods, in: AIAA/AAS Astrodynamics Specialist Conference, 2010, pp. 2–5.
[66] C.L. Darby, W.W. Hager, A.V. Rao, Direct trajectory optimization using a variable low-order adaptive pseudospectral method, J. Spacecr. Rockets 48 (2011) 433–445.
[67] C.L. Darby, W.W. Hager, A.V. Rao, An hp-adaptive pseudospectral method for solving optimal control problems, Optim. Control Appl. Methods 32 (2011) 476–502.
[68] F. Liu, W.W. Hager, A.V. Rao, Adaptive mesh refinement method for optimal control using nonsmoothness detection and mesh size reduction, J. Franklin Inst. 352 (2015) 4081–4106.
[69] D.A. Benson, G.T. Huntington, T.P. Thorvaldsen, A.V. Rao, Direct trajectory optimization and costate estimation via an orthogonal collocation method, J. Guid. Control Dyn. 29 (6) (2006) 1435–1440.
[70] H. Liu, S. Chen, L. Shen, J. Chen, An integrated multicriterion hp-adaptive pseudospectral method for direct optimal control problems solving, Math. Probl. Eng. (2012).
[71] A.M. Yazdani, K. Sammut, A. Lammas, B. Clement, O. Yakimenko, Cooperative guidance system for AUV docking with an active suspended docking station, Oceans 2019 (2019) 1–7.
[72] A.M. Yazdani, K. Sammut, A. Lammas, Y. Tang, Real-time quasi-optimal trajectory planning for autonomous underwater docking, in: IEEE International Symposium on Robotics and Intelligent Sensors, IRIS, 2015, pp. 15–20.
[73] A.M. Yazdani, K. Sammut, O. Yakimenko, A. Lammas, S.M. Zadeh, Y. Tang, Time and energy efficient trajectory generator for autonomous underwater vehicle docking operations, Oceans 2016 (2016) 1–7.
[74] A.M. Yazdani, K. Sammut, O. Yakimenko, A. Lammas, Launch and recovery for ship-deployed autonomous underwater vehicles, in: International Conference on Ship and Maritime Research, 2018, pp. 596–603.
[75] D.S. Naidu, Optimal Control Systems, CRC press, ISBN: 978-084-9308-92-5, 2002.

Amir Mehdi Yazdani received the master's degree in control engineering from Universiti Teknologi Malaysia, Johor Bahru, Malaysia, in 2012, and the Ph.D. degree in control engineering from Flinders University, Adelaide, SA, Australia, in 2017. He was employed between 2017 and 2018 as a Postdoctoral Research Associate with Flinders University. In 2019, he joined the Murdoch University, WA, Australia and he is currently working as a Lecturer in Electrical Engineering in the Engineering and Energy Discipline. His current research interests include guidance and control of unmanned vehicles, robotic and mechatronic systems, optimal and nonlinear control, system management and optimization, and intelligent control applications.

Karl Sammut completed his Ph.D. at The University of Nottingham (U.K) in 1992 and was employed between 1992 and 1995 as a Postdoctoral Fellow with the Politecnico di Milano (Italy), and with Loughborough University (UK). He commenced his appointment at Flinders University in 1995 and is currently a Professor with the Engineering Discipline in the College of Science and Engineering at Flinders University. He serves as the Director of the Centre for Maritime Engineering at Flinders University and the Theme Leader for the Maritime Autonomy Group. His areas of research specialization are concerned with navigation, optimal guidance and control systems, and mission planning systems for autonomous marine surface and underwater vehicles.

Oleg Yakimenko is a Distinguished Professor of Systems Engineering and Mechanical and Aerospace Engineering at the Naval Postgraduate School, Monterey, CA. He currently serves as an Associate Dean of Research. He holds two doctoral degrees — in Aerospace Engineering (1991) and in Operations Research (1996). He serves as the Programme Manager of the Autonomous Systems track, Director of the Aerodynamic Decelerator Systems Center, and Director of Autonomous Systems Engineering and Integration Laboratory. His areas of interest include modelling, guidance, navigation and control of manned and unmanned aerial and maritime vehicles, satellites, guided weapons and parachutes. Dr. Yakimenko is an author or co-author of over 300 publications including 14 textbooks. He is Associate Fellow of the American Institute of Aeronautics and Astronautics and Fellow of the Russian Academy of Sciences of Aviation and Aeronautics.

Andrew Lammas received his Bachelor of Engineering (Computer Systems) from Flinders University, Adelaide, Australia in 2004. He completed his Ph.D. in Engineering also at Flinders University in 2012 focusing on underwater navigation. His research interests include design and implementation of model based robust nonlinear filtering methods for navigation and awareness systems within autonomous vehicles. He is currently a research associate within the Centre of Maritime Engineering, Control, and Imaging (CMECI) in the College of Science and Engineering at Flinders University. In his current role he is working as a postdoctoral researcher with the Centre for Maritime Engineering on the development of localization and mapping systems for automated inspection robots operating in confined spaces, as well as on battery pack modelling for large scale energy storage systems.